OPERATIONAL PRIORITIES (OPSPRI) MODEL VERSION 1.1: USERS MANUAL

Report AF001R1

May 1991

Tovey C. Bachman

Prepared pursuant to Department of Defense Contract MDA903-90-C-0006. The views expressed here are those of the Logistics Management Institute at the time of issue but not necessarily those of the Department of Defense.

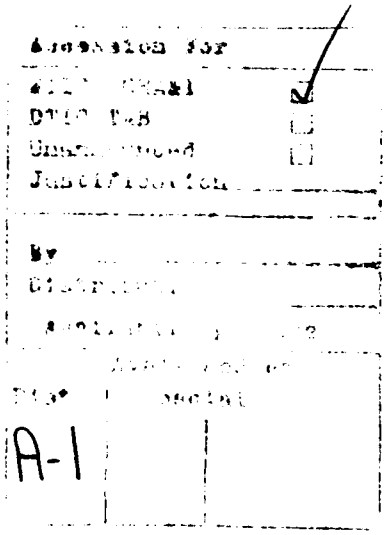
LOGISTICS MANAGEMENT INSTITUTE
6400 Goldsboro Road
Bethesda, Maryland 20817-5886

PREFACE

This users manual is divided into eight chapters and four appendices. We present an overview of the model in Chapter 1, the hardware and software requirements in Chapter 2, and the input files and preprocessors descriptions in Chapter 3. Chapters 4, 5, and 6 explain model installation, operation, and its use in providing policy guidance. Chapters 7 and 8 describe the algorithms used in the model and the troubleshooting procedures. In the appendices, we show input file formats (Appendix A), output file format (Appendix B), explain relative availability (Appendix C), and provide the glossary of terms used in this report (Appendix D).

Those readers who wish to become quickly acquainted with the model should immediately turn to Chapters 4 and 5. The program diskettes contain a sample data base.

This users manual refers to the Operational Priorities (OPSPRI) model Version 1.1. Version 1.0, distributed for demonstration, uses different file formats and does not incorporate output reports or weighted average availability.



Executive Summary

OPERATIONAL PRIORITIES (OPSPRI) MODEL VERSION 1.1: USERS MANUAL

Headquarters Air Force's Operations/Logistics Working Group has developed a scheme for allocating logistics resources to units based on their operational priorities. This method ensures that a unit's level of support reflects the criticality of its mission but retains balanced support across the entire force. The OPSPRI model implements this scheme for allocation of war readiness spares kit/base level self-sufficiency spares funding and provides several measures of the resulting capability.

The model uses aircraft availability curves from the Air Force's Weapon System Management Information System (WSMIS) to relate support levels to the cost of achieving those support levels. It uses implied turn rates from WSMIS flying-hour scenarios to estimate available sorties.

In performing a funding allocation, the user may adjust funds for a particular priority group, theater, major command, mission design, or unit. Funding may be increased, decreased, or eliminated selectively. Once funds are allocated, the OPSPRI model displays the results of the decisions from a variety of perspectives. For example, if the initial funding decision is made by priority group/theater, the user can view the resulting capability by mission design. After the model is run, a file showing funding allocation is returned to the Air Force Logistics Command for use in determining spares buys. This manual explains the model, its data requirements, and the concepts behind it.

CONTENTS

CONTENTS (Continued)

FIGURES

FIGURES (Continued)

CHAPTER 1

OVERVIEW

INTRODUCTION

The Operational Priorities (OPSPRI) model enables the user to allocate funds for war readiness spares kits (WRSK) and base level self-sufficiency spares (BLSS) across Air Force units in accordance with operational guidance. The resulting capability may then be viewed by priority group, theater, priority group/theater, major command (MAJCOM), mission design (MD), or unit.

This chapter briefly describes the model, its uses, and its relation to other Air Force data systems.

MODEL DESCRIPTION

The OPSPRI model is a personal computer (PC)-based, interactive model that immediately displays results, permitting the user to quickly perform "what if" analyses as well as budget allocation for a single funding level. The display format is a matrix in which the rows are groups of units [e.g., the second row could be European Command (EUCOM) units] and the columns are levels of support. Support is measured by relative availability, or the availability relative to the requirement. Appendix C provides for a thorough explanation of relative availability.

Funds are allocated through the following procedure developed by the Air Force Operations/Logistics (Ops/Log) Working Group: Each unit has Support Levels A, B, C, etc., expressed as a percentage of required capability. Funds are first allocated to bring units up to Support Level A, with units receiving funding in the order determined by their priorities. If sufficient funds are available, all units will be supported to Level A. Funds are then allocated to bring units up to Support Level B, and then Level C, and so on; for each support level, units receive funds in priority order, just as they did for Level A. The process continues until all funds are exhausted. This allocation method ensures that higher priority units are better supported while maintaining some balance in support across all units.

If funds run out while bringing units up to, say, Support Level H, then some units will be supported at Level H, others will only be supported at Level G, and the last unit to receive funds may be supported beyond Level G without attaining level H. In Figure 1-1, WARFIGHTING/SOUTHCOM, WARFIGHTING/TRANSCOM, and SUPPORT/EUCOM are all fully funded through Level H, as are priority groups/ theaters above them (not visible). SUPPORT/LANTCOM is fully funded through Level G and partially funded through Level H. No priority groups/theaters below SUPPORT/LANTCOM are funded beyond Level G.

HELP-F1 EXIT-ESC

OPS Priorities--AGGREGATED BY PRIORITY GROUP/THEATER

Row 18
Col A

Priority Group/Theater	Availability / $ Millions						Current Decision	
	Support Levels G		H		I			
WARFIGHTING/SOUTHCOM	[illegible]	[illegible]	[illegible]	[illegible]	72%/$	1	70%/$	1
WARFIGHTING/TRANSCOM	[illegible]	[illegible]	[illegible]	[illegible]	72%/$	50	70%/$	44
SUPPORT/EUCOM	[illegible]	[illegible]	[illegible]	[illegible]	72%/$	15	70%/$	13
SUPPORT/LANTCOM	[illegible]	[illegible]	70%/$	39	72%/$	45	69%/$	36
SUPPORT/PACOM	[illegible]	[illegible]	70%/$	14	72%/$	16	68%/$	12
SUPPORT/CENTCOM	[illegible]	[illegible]	70%/$	56	72%/$	63	68%/$	49
SUPPORT/ALASKCOM	[illegible]	[illegible]	70%/$	14	72%/$	16	68%/$	12
Average / Total	68%/$	456	70%/$	520	72%/$	587	69%/$	500

= fully funded (blue) = partially funded (green)

Notes: OPS = operational; SOUTHCOM = Southern Command; TRANSCOM = Transportation Command (strategic airlift and tankers); LANTCOM = Atlantic Command; PACOM = Pacific Command; CENTCOM = Central Command; ALASKCOM = Alaskan Command.

FIG. 1-1. OPERATIONAL PRIORITIES MATRIX FULLY FUNDED THROUGH LEVEL G AND PARTIALLY FUNDED THROUGH LEVEL H

The funds required for each unit to achieve a support level (i.e., a level of capability) are computed from aircraft availability curves, a process described in Chapter 7.

The user can override this OPSPRI allocation process by adjusting the funds for a particular priority group, theater, priority group/theater, MAJCOM, MD, or unit. Funding for this chosen group of units may be increased, decreased, or eliminated

without affecting funding of other units. If constant total funding is desired, the user will, of course, need to decrease funding of some units to increase that of others.

Once funds have been allocated, the user may view the results of the decision from a variety of perspectives. For example, if the initial funding decision were made by priority group/theater, the user could view the resulting capability by MD. Saving the results of an OPSPRI session is simple and takes less than half a minute on most PCs.

The OPSPRI model also estimates sorties and flying hours resulting from the allocation. It displays results by plotting a graph on the screen, sending a table to a text file, or sending a table to a printer. When a final funding decision has been reached, OPSPRI produces an output file, showing dollars allocated by unit, to be returned to the Air Force Logistics Command (AFLC). AFLC will convert that dollars-by-unit allocation to a dollars-by-item allocation for WRSK/BLSS procurement.

USES OF THE MODEL

As noted, we can use OPSPRI to examine the capability of groups of units (e.g., theaters) or individual units obtained with a particular level of funding, allocated according to the operational priorities concept. Further, the model can also be used to examine the following conditions for a unit or group of units:

- Sensitivity of capability to changes in funding
- Changes in capability relative to aircraft availability requirements or to required funding resulting from changes in requirements
- Changes in capability or required funding resulting from changes in force structure
- Changes in capability or required funding resulting from changes in the relative importance of theaters.

We discuss uses of the model in greater detail in Chapter 5.

RELATION TO OTHER AIR FORCE DATA SYSTEMS

The OPSPRI model has interfaces with other Air Force data systems for both input and output. Some required data are not currently provided by automated data systems. The Ops/Log Working Group's operational priority matrix, which associates

each priority group/theater combination with a set of support levels, must be supplied to the model. That matrix is not part of an automated data system.

For each WRSK/BLSS buy kit, the Requirements/Execution Availability Logistics Module (REALM) of the Weapon System Management Information System (WSMIS) provides a pair of curves relating dollars to aircraft sustainability, one for a primary day of the scenario and another (optional) for a secondary priority day. Here, "buy kit" refers to a WRSK whose composition is used for procurement purposes. Actual kits received by units, known as "contingency kits," may differ from the buy kit on which they are based. For each kit, WSMIS/REALM also supplies a scenario file containing flying hours by day.

The Automated War and Mobilization Plan, Volume 3 (Automated WMP-3) provides theater, priority group, MAJCOM, mission design series (MDS), and primary aircraft authorized (PAA) for each unit; in some cases, this information is also sufficient to determine the buy kit for that unit. WMP-5 supplies sortie length by MDS.

For an MDS with unit-specific buy kits, it may be necessary to contact the WRSK system program manager (SPM) for the weapon system to match units and kits. In Chapter 3, we give a detailed explanation of the file UNITKIT.DAT, which shows the unit-kit correspondence.

After funding allocation is complete, OPSPRI returns the dollar amount for each WRSK/BLSS buy kit by unit to WSMIS/REALM for budget execution.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE

The OPSPRI model requires an IBM-compatible PC with an 80286, 80386, or 80486 Intel processor and a hard disk drive; an IBM AT or a Zenith 248, for example, are satisfactory. The model may run on a machine with an 8086 or 8088 processor, (e.g., IBM XT), but it will be extremely slow. An 80287 or 80387 math coprocessor is not necessary.

To perform a budget allocation for 400 units, you need about 2 megabytes of hard disk space for data files. The OPSPRI executable file consumes only 100K bytes and the data base files require approximately 650K bytes. Your machine should have at least 350K of random access memory (RAM) available after the disk operating system (DOS) and any other memory-resident programs have been loaded. The amount of RAM and hard disk space needed are roughly proportional to the number of kits; for 500 units, you will need about 450K of free RAM and 3.5 megabytes of hard disk space. These figures are estimates; actual RAM and hard disk space requirements are a function not only of the number of units, but also of the distribution of units across priority groups, theaters, etc.

Your monitor and video card should be either video graphics array (VGA), enhanced graphics adapter (EGA), color graphics adapter (CGA), or a monochrome monitor with a Hercules graphics card. Some display features will not work on strict monochrome (no graphics) monitors.

SOFTWARE

The model will run on a computer with DOS Version 2.0 or higher. Your CONFIG.SYS file should include the command FILES=20.

CHAPTER 3

INPUT FILES AND PREPROCESSORS

The OPSPRI model uses some input files directly, while others are built with preprocessor programs. In this chapter, we describe directly used input files first.

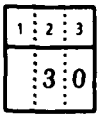
DIRECTLY USED INPUT FILES

The directly used input files include list files, scenario files, or graphics files for plotting. The list files are THTRLIST.DAT, which lists theaters in order of priority, where "higher priority" means "higher on the list"; PRGPLIST.DAT, which lists priority groups in order of priority; CMNDLIST.DAT, which lists MAJCOMs in priority order if such a priority exists; and MDLIST.DAT which lists mission design in priority order if such a priority exists.

The order of items in each list file is important because it is used to determine the order in which the items receive funds if the user elects to perform the allocation by that type of item. For example, if the user chooses to allocate funds by theater, theaters will receive increments of funding in the order in which they are listed, starting at the top of the list.

Each file has one header line that can be used to label the file; that line is ignored by OPSPRI. Theater names may be up to 9 characters long, priority groups up to 12 characters, MAJCOMs up to 8 characters, and MDs up to 5 characters. All names must be left-justified and no blank lines may be included except possibly the header line.

The OPSPRI model requires a scenario file for each unit. Columns 1–3 of the first line contain the number of days in the scenario. That number should be right-justified: for example, *30 days* should be entered as



Subsequent rows contain flying hours for the unit in Columns 1 – 8, with a decimal point in Column 6.

The graphics files are provided by Borland's Turbo Pascal 5.5. These files, listed below, must be in the same directory as OPSPRI.EXE:

- ATT.BGI
- CGA.BGI
- EGAVGA.BGI
- HERC.BGI
- IBM8514.BGI
- PC3270.BGI
- SANS.CHR
- TRIP.CHR.

The OPSPRI model reads these files when it plots results on the screen.

PREPROCESSORS

In this section, we describe the preprocessors UNITOCRV, OPSMERGE, and MAKECRVS and their associated input files. These preprocessors supply OPSPRI with the input files PRIORITY.DAT, CROSSREF.DAT, and the relative availability curve files.

If the data are to be preprocessed on a PC, copy the executable file (e.g., UNITOCRV.EXE) and the appropriate input files to your working directory. If the data are to be preprocessed on a mainframe computer or minicomputer, the preprocessor source code (e.g., UNITOCRV.FOR), written in Lahey FORTRAN 77, must first be compiled and linked to produce executable code that is compatible with your computer/operating system. While the source code was written with portability in mind, you may need to edit it to enable it to compile, link, and run in your environment. The resulting executable file must be in the same directory as the input files before running the preprocessor.

MAKECRVS Preprocessor

The MAKECRVS preprocessor produces the relative availability curve files that OPSPRI uses to determine the cost of achieving the support levels contained in PRIORITY.DAT. To run MAKECRVS, you must have the file CRVFILES.DAT, one or two REALM aircraft availability curve files for each unit, and the executable file for MAKECRVS in your working directory. The preprocessor will warn you if an output curve file already exists. Move the existing output curve files to a new directory before running MAKECRVS.

The input file CRVFILES.DAT has the following format: The first line is a header line and is ignored by MAKECRVS. On all subsequent lines, Columns 1 – 40 contain the path and file name for the availability curve for the primary day, and Columns 41 – 80 contain the path and file name for the aircraft availability curve for the secondary day, if any. (The section on "Generating Reports" in Chapter 7 gives an explanation of primary and secondary days.) All path and file names should be entered as left-justified character strings:

41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
D	:	\	S	O	U	T	H	C	O	M	\	T	A	C	\	F	1	6	\	C	U	R	V	E	.	C	R	V											

Columns 81 – 83 contain the primary day, Column 84 is blank, and Columns 85 – 87 contain the secondary day, if any. Days should be right-justified: for example, if *Day 7* is the primary day, it should be entered as

85	86	87
		7

Curve file names for the primary day must use the extension *.CR1*. Curve file names for the secondary day (if present) must use the extension *.CR2*. The relative availability curve file name will be the same as the primary day curve file name except for its extension, which will be *.CRV*.

Each input aircraft availability curve must have the following format: In the first line, Columns 1 – 12 must contain the first 12 characters of a kit serial number (KSN). Next must come an integer PAA, spaces, and a direct support objective (DSO). These fields are space delimited; the exact columns for these numbers are not important. The second line is a header (comment) of up to 80 characters that will be

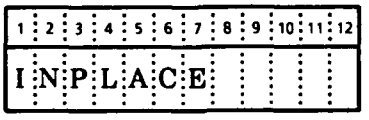
copied into the output curve file. For all subsequent lines, Columns 1–3 must contain blanks, Columns 4–10 must contain expected not mission capable-supply (ENMCS) aircraft, Columns 11–26 are ignored by MAKECRVS, Columns 27–36 must contain cost, and the remaining columns are ignored by MAKECRVS. The line after the line with the smallest positive ENMCS and the largest cost must be one with ENMCS = 0.000 and cost = 0.0. That line is used to indicate the end of the ENMCS-versus-cost curve; all data after that line are disregarded by MAKECRVS.

On a PC, you may run MAKECRVS by simply typing "MAKECRVS" at the DOS prompt and pressing *Enter*. When the program is finished, you should see a summary of results on your screen or system printer and the words *PROCESSING COMPLETE*.

OPSMERGE Preprocessor

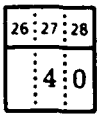
This preprocessor produces the file PRIORITY.DAT, which OPSPRI uses to match each unit (or subset of a unit combination) with its MD, priority group, theater, MAJCOM, and support levels. To run OPSMERGE, the input files OPSLOG.DAT and WMP3.DAT must be in the directory in which you wish PRIORITY.DAT to be written. The preprocessor will warn you if the file PRIORITY.DAT already exists. Move the existing PRIORITY.DAT file to a new directory before running OPSMERGE.

The input file OPSLOG.DAT is a text file with the following format: The first two lines are header lines and are ignored by OPSMERGE. For all subsequent lines, Columns 1–12 contain a priority group. If that priority group is less than 12 characters, it should be left-justified: for example, *INPLACE* should be entered as



Columns 13–14 are blank. Columns 15–23 contain a theater. If the theater is less than 9 characters, it should be left-justified. Columns 24–25 are blank. Columns 26–28 contain a three-digit integer representing Support Level A. If the

support level is less than 3 characters, it should be right-justified: for example, *40* should be entered as



Column 29 is blank, Columns 30–32 contain an integer representing Support Level B, Column 33 is blank, Columns 34–36 contain Support Level C, and so on. Currently, the maximum number of support levels is 26, with Support Level Z in Columns 126–128. If the unit has fewer than 26 support levels, all columns following the column containing the last support level must be left blank. Each row in OPSLOG.DAT *must have the same number of support levels.*

The input file WMP3.DAT is a text file with the following format: The first two lines are header lines and are ignored by OPSMERGE. For all subsequent lines, Columns 1–13 contain a unit name, left-justified; Columns 14–15 are blank; Columns 16–18 contain a subunit designator (usually a squadron/wing detachment – a subset of a unit with distinct tasking); Columns 19–20 are blank; Columns 21–29 contain a theater, left-justified; Columns 30–31 contain blanks; Columns 32–43 contain a priority group, left-justified; Columns 44–45 are blank; Columns 46–53 contain a MAJCOM, left-justified; Columns 54–55 are blank; Columns 56–61 contain an MDS, right-justified; Columns 62–63 are blank; Columns 64–67 contain sortie length, right-justified; and Columns 68–69 contain blanks.

Columns 70–72 of the WMP3.DAT input file contain a PAA, which is not used directly by OPSPRI but is present because MAJCOM, MDS, and PAA are needed to determine the KSN corresponding to this unit (or subset of a unit). This correspondence is shown in the file UNITKIT.DAT.

On a PC, you may run OPSMERGE by simply typing "OPSMERGE" at the DOS prompt and pressing *Enter*. When the program is completed, you should see a summary of results on your screen or system printer and the words *PROCESSING COMPLETE*. Copy the file PRIORITY.DAT to the directory on the PC on which you wish to install OPSPRI.

UNITOCRV Processor

The UNITOCRV preprocessor produces the file CROSSREF.DAT, which the OPSPRI model uses to match units with their relative availability curve files and scenario files. To run UNITOCRV on a PC, UNITOCRV.EXE and the two input files UNITKIT.DAT and KITFILE.DAT must be in the directory to which CROSSREF.DAT is to be written. On other computers, the executable file for UNITOCRV must be in the same directory as the input files. The two input files are described below. The preprocessor will warn you if the file CROSSREF.DAT already exists. Move the existing CROSSREF.DAT file to a new directory before running the preprocessor.

The file UNITKIT.DAT is a text file with the following format: Lines 1 – 2 are header lines and are ignored by UNITOCRV. For all subsequent lines, Columns 1 – 13 contain the unit identifier. If the unit identifier is less than 13 characters, it should be left-justified: for example, *125th TFS* (Tactical Fighter Squadron) should be entered as

1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	5	t	h	T	F	S					

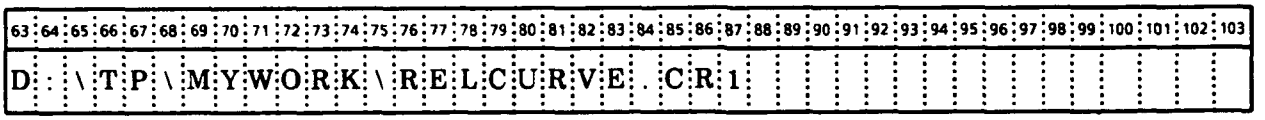
Columns 14 – 15 are blank. Columns 16 – 18 contain SUB, a character code for a squadron/wing detachment, a subgroup of a unit that may be given distinct tasking. SUB should be right-justified: for example, if SUB is *a*, it should be entered as

16	17	18
		a

Note that each row in the file may be a subset of a unit, or two units being treated as one. It is crucial that there be a REALM curve file corresponding to each row, whether or not that row is a single (entire) unit. The same unit identifier may appear in more than one row. For example, that would be the case if a unit has more than one aircraft type. Columns 19 – 20 are blank, and Columns 21 – 32 contain the first 12 characters of a KSN.

The file KITFILE.DAT is a text file with the following format: Lines 1 – 2 are header lines and are ignored by UNITOCRV. For all subsequent lines, Columns 1 – 12 contain the first 12 characters of a KSN, Columns 13 – 20 are blank,

Columns 21 – 60 contain the REALM curve path and file name for the primary day of the scenario, Columns 61 – 62 are blank, and Columns 63 – 103 contain the scenario file path and file name. Path and file name must be entered as left-justified character strings:



The primary day curve file names are used to produce relative availability curve file names for the CROSSREF.DAT file.

On a PC, you may run the UNITOCRV preprocessor by simply typing "UNITOCRV" at the DOS prompt and pressing *Enter*. On other computers, you may need to go through a procedure specific to your operating system in order to run the program.

When the program is completed, you should see a summary of results on your screen or system printer along with the words *PROCESSING COMPLETE*. Copy the file CROSSREF.DAT to the directory on your PC on which you wish to install OPSPRI.

CHAPTER 4

INSTALLATION

Before installing OPSPRI on your PC, verify that your hardware and software configuration meets the conditions specified in Chapter 2. The first part of this chapter lists the files needed and explains how to check them for compatibility. The second part explains file installation.

CHECKING FILES

First, check that you have the following necessary files:

- *OPSPRI.EXE* – contains the OPSPRI program
- *Relative availability curve files* – one for each unit
- *Scenario files* – one for each unit
- *CROSSREF.DAT* – provides curve and scenario file names for each unit
- *PRIORITY.DAT* – contains MD, MAJCOM, theater, priority group, and support levels for each unit
- *MDLIST.DAT* – ordered list of MDs
- *CMNDLIST.DAT* – ordered list of MAJCOMs
- *PRGPLIST.DAT* – ordered list of priority groups
- *THTRLIST.DAT* – ordered list of theaters
- *Graphics files:*
  - ATT.BGI
  - CGA.BGI
  - EGAVGA.BGI
  - HERC.BGI
  - IBM8514.BGI
  - PC3270.BGI

- SANS.CHR
- TRIP.CHR.

You must have a relative availability curve file (created by MAKECRVS) for every file name that is listed in CROSSREF.DAT. If there is no curve file corresponding to a particular file name, OPSPRI will stop and display an error message. You will need either to provide the required file before rerunning the model or, if the file name in CROSSREF.DAT is in error, to edit CROSSREF.DAT so that it includes the correct file name. Similarly, you must have a scenario file for every scenario file name listed in CROSSREF.DAT.

Each unit's MD, shown in PRIORITY.DAT, must be on the MD list in MDLIST.DAT. Theaters, priority groups, and MAJCOMs for units in PRIORITY.DAT must also be in their corresponding list files. If they are not, either the missing item must be added to the list (e.g., to the file MDLIST.DAT, if the missing item is an MD) or PRIORITY.DAT must be edited. If a discrepancy of this type occurs, OPSPRI will stop and display an error message.

All items must be listed in order of priority, with the highest priority item first. For example, INPLACE is a higher priority group than DETERRENT, so it should appear before DETERRENT on the list. If two items have equal priority or if no ranking is meaningful, they may be listed in either order.

INSTALLING FILES

Create a subdirectory on your hard disk called OPSPRI and enter all the files (*.*) from Diskettes 1 and 2 into this directory. These diskettes include the OPSPRI model and a sample data base. Put all files from Diskettes 3 and 4 into a separate directory, PREOPS. These diskettes contain the preprocessor programs and sample input files for those programs.

All of the sample data files for the model are now in the OPSPRI directory. (Preprocessor data are in the PREOPS directory.) With real data, curve files and scenario files (extensions *.CR1*, *.CR2*, *.CRV*, and *.SCN*) may be in another directory as long as the CRVFILES.DAT and KITFILES.DAT files specify the correct path/file names.

If you copy all input files for the model into the root directory (e.g., C:), you may exceed the maximum number of files allowed (512) under DOS Versions 2.0–4.0. This problem will *not* occur as long as files are copied into a subdirectory (e.g., C:\OPSPRI).

CHAPTER 5

OPERATION

STARTING THE MODEL

Change directories if necessary so that the DOS prompt shows that you are in the OPSPRI subdirectory. Type "OPSPRI" followed by a return (*Enter key*).

The first time you run the model, you should see a message that tells you that a disk data base is being created (see Figure 5-1). If you receive an error message instead, check file compatibility (Chapter 4) and file format (Appendix A). After several minutes, you will see another message, telling you that the screen display is being created (see Figure 5-2). Several seconds later, you should see a matrix display, shown in Figure 5-3.

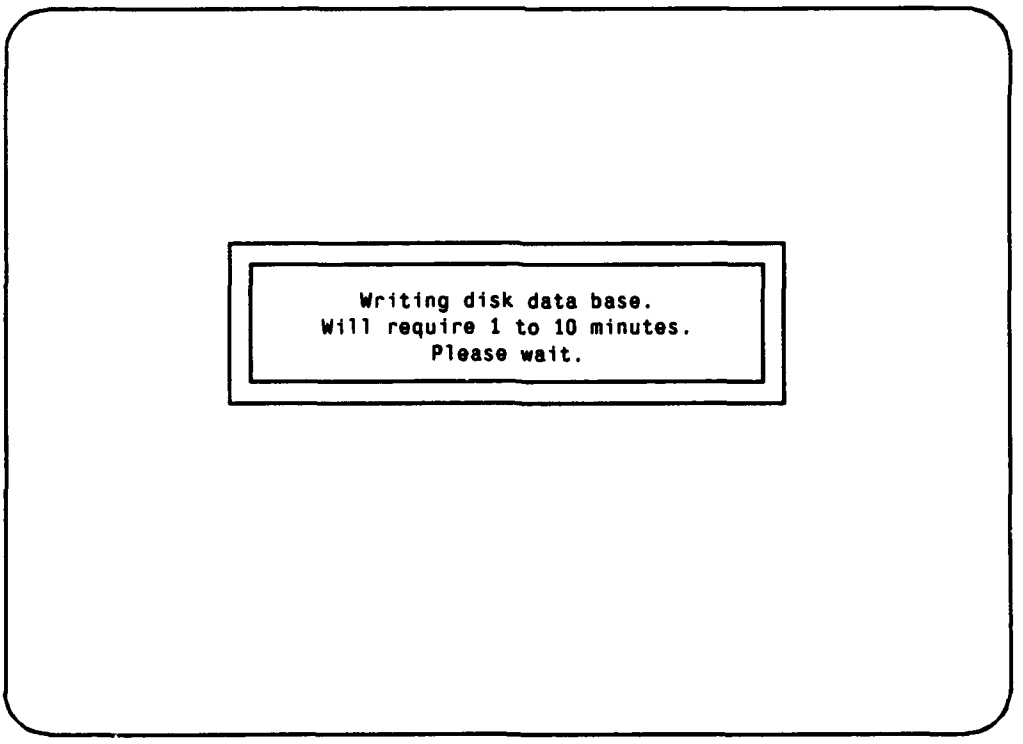


FIG. 5-1. SCREEN DISPLAY WHILE OPSPRI IS CREATING DISK DATA BASE

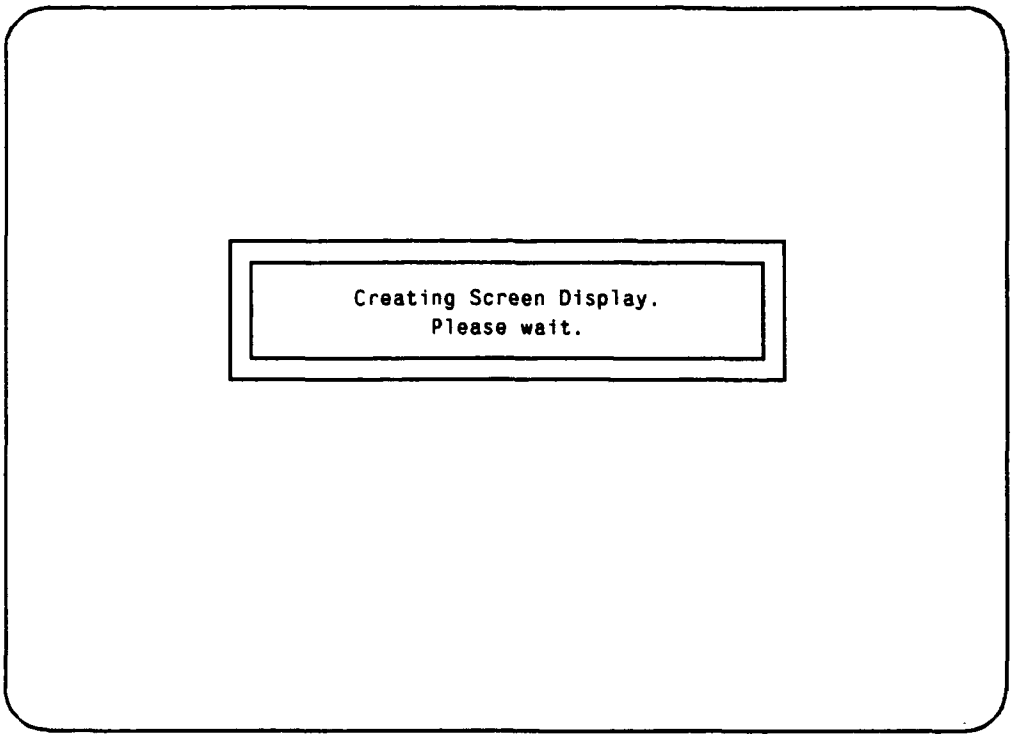


FIG. 5-2. SCREEN DISPLAY WHILE OPSPRI IS CREATING SCREEN DATA BASES

HELP-F1 EXIT-ESC

OPS Priorities--AGGREGATED BY PRIORITY GROUP/THEATER

Priority Group/Theater	Availability / $ Millions						Row 1 Col 0
	Support Levels						Current Decision
	0		A		B		
INPLACE/NORAD	20%/$	0	28%/$	1	32%/$	4	20%/$ 0
INPLACE/EUCOM	24%/$	0	28%/$	2	32%/$	13	24%/$ 0
INPLACE/LANTCOM	23%/$	0	28%/$	0	32%/$	2	23%/$ 0
INPLACE/PACOM	24%/$	0	28%/$	2	32%/$	15	24%/$ 0
INPLACE/SOUTHCOM	20%/$	0	28%/$	1	32%/$	4	20%/$ 0
DETERRENT/LANTCOM	24%/$	0	28%/$	2	32%/$	10	24%/$ 0
DETERRENT/CENTCOM	23%/$	0	26%/$	0	28%/$	1	23%/$ 0
Average / Total	23%/$	0	26%/$	9	29%/$	60	23%/$ 0

Note: NORAD = North American Air Defense.

FIG. 5-3. INITIAL MATRIX DISPLAY

If the disk data base has already been created, the user sees a menu offering a choice between re-creating the data base or using the existing data base (see Figure 5-4). Choose *NO* – do not re-create the disk data base – if relative availability curves, support levels, *or* other input data have not changed; simply press *Enter*. You may also type "n", or "N". Choose *YES* if there has been a change in the input data and you would like to update the data base. When a real data base is brought in to replace the sample data base, you must choose *YES*. *Warning*: Choosing *YES* causes your old data base to be overwritten. If you wish to save the old data base, exit the model with *Control-Break* (*Ctrl-Break*), and either copy the old data files to another subdirectory or rename them. Now reenter OPSPRI and select *YES*. To select yes, use the *down arrow key*, followed by *Enter*. You may also type "y", or "Y".

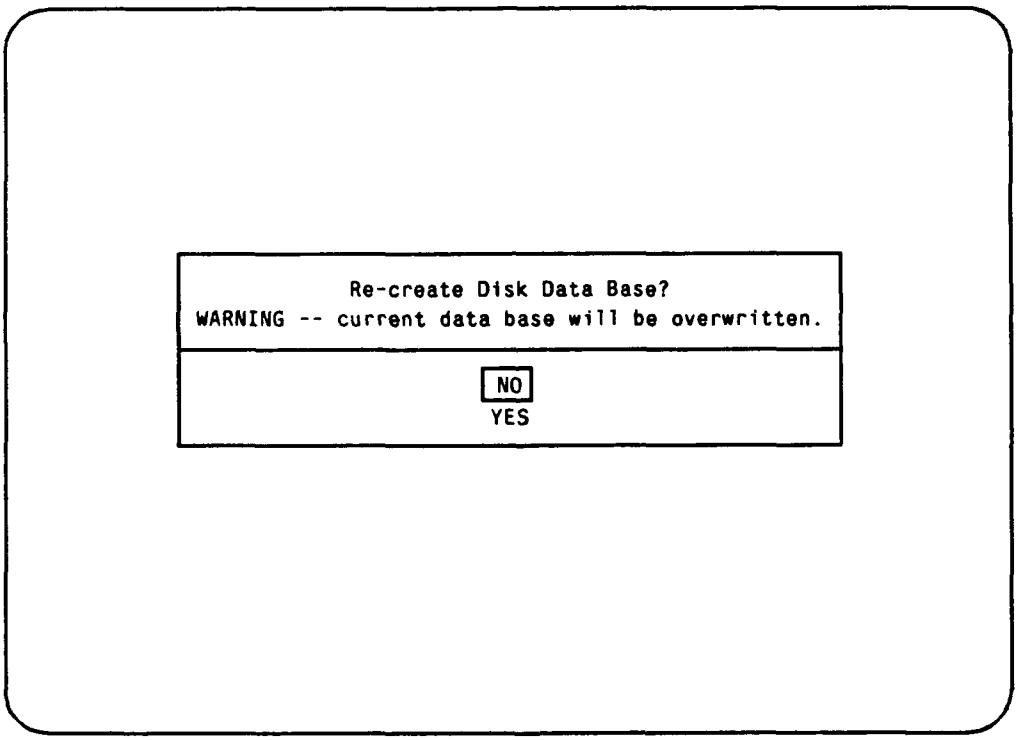


FIG. 5-4. DISK DATA BASE CREATION MENU

If you select *NO*, you will see the message *Creating Screen Display*. After several seconds, you will see the initial operational priorities matrix display, shown in Figure 5-3. If you select *YES*, you will see the message telling you that the disk data base is being created. Several minutes later, you will see the message stating

that the screen display is being created, followed by the matrix display shown in Figure 5-3.

OVERVIEW OF DISPLAY

In the initial matrix display in Figure 5-3, the user is looking through a window on the operational priorities matrix. Only Columns 0, A, and B and the first seven rows of the matrix are visible (the Current Decision column and the Average/Total row are not inside the window.) By using the *arrow keys*, the user can scroll right to see columns beyond B, or down to see rows below Row 7. The *F1 key* brings up the help screen, which is a summary of all the key functions.

The "current cell" is indicated by cell entries in yellow or green; all other cells have entries in white. Each time an arrow key is pressed, the current cell moves in the direction of the arrow by one cell. A box in the upper right-hand corner displays the row and column of the current cell. The *End key* moves the current cell to the last column, and the *Home key* brings the current cell back to the first column. *Ctrl-Home* brings the current cell to the top row, and *Ctrl-End* brings it to the bottom row (not Average/Total). *PgDn* moves the current cell down six rows, and *PgUp* moves it up six rows.

Initially, each row except the last one represents a priority group/theater combination, such as INPLACE/LANTCOM. Columns represent support levels for the priority groups/theaters. Each cell contains a percentage relative availability and the cost of achieving that availability, measured in millions of dollars. The availabilities are support levels from the Ops/Log Working Group priority matrix. Costs are computed from relative availability curves. A discussion of these curves is presented in Appendix C. Except for cells in the Current Decision column, neither the support levels nor the costs of any cells will change during an OPSPRI session. Numbers in cells outside the Current Decision column will change only if the data base is re-created using a different operational priorities matrix or using different aircraft availability curves.

Cells in the last row, Average/Total, show the weighted average availability obtained by bringing all priority groups/theaters to Support Levels A, B, C, and so on, and the total cost of bringing all priority groups to each support level. For example, in Figure 5-3, the average availability when all priority groups/theaters are funded to Level A is 26 percent, and the cost is $9 million. Weighted average availability is

computed by summing the availabilities of units, weighted by their PAAs, across all units and then dividing by the sum of the PAAs.

Cells in the Current Decision column show the dollars spent on each priority group/theater with current total funding and the resulting availability. For example, in Figure 5-3, where we have not yet added any funding to the matrix, the availability for INPLACE/NORAD is 20 percent and the dollars allocated to INPLACE/NORAD is zero. The availability is not zero because even a unit with an empty spares kit will usually have some aircraft that can fly each day of the scenario. The cell in the bottom right-hand corner shows current total funding and the resulting weighted average availability for all priority groups/theaters.

ALLOCATION OF FUNDS

To add funds, press the *+ key*. Each keystroke adds $1 million to the total. A cell is black if the priority group/theater in its row has not been funded to the support level in its column; it is green if the priority/group theater is partially funded through the support level; and it is blue if the priority group theater is fully funded through the support level. Funds are allocated by the Ops/Log Working Group method in which each priority group/theater first receives funds to bring it to Support Level A, starting with the one at the top of the matrix and working down the first column. When all priority groups/theaters are funded through Level A, the OPSPRI model begins allocating funds to bring priority groups/theaters to Level B, again starting at the top of the column, and so on. Allocation stops when funding is exhausted. Note that the cells in the Current Decision column change as funds are added to reflect the current expenditure on each priority group/theater and the resulting availability. To remove funds, use the *– key*. Dollars are removed in the reverse order of the way in which they were allocated.

To change the size of the dollar increment, press *F4*. You will now see the menu shown in Figure 5-5. Enter dollars in millions, using a decimal point if you wish. For example, if you wish to add funds to the matrix in increments of $1,250,000, you would enter "1.25". Now press *Enter*. Subsequent use of the *+* and *– keys* will use 1.25 million as the amount of dollars to be added or subtracted for each keystroke, unless you use *F4* to change it. Increment size is not saved when you leave an OPSPRI session.

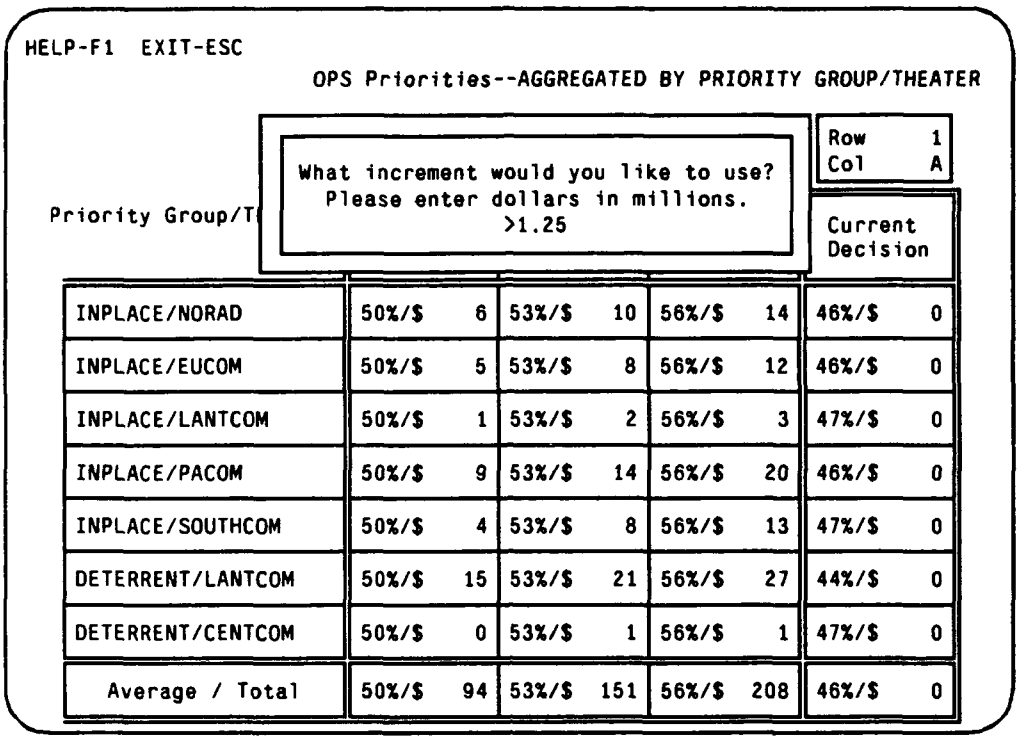


FIG. 5-5. FUNDING INCREMENT MENU (*F4*)

To go directly to a funding level, press *F3*. Enter dollars in millions and press *Enter*. The bottom right-hand corner cell now shows the amount of funds added and the resulting weighted average availability. The other Current Decision cells show the amount spent on each priority group/theater and the resulting availability. For example, in Figure 5-6, we have decided to spend $350 million. Figure 5-7 shows the result of this decision – we have a weighted average availability of 63 percent, INPLACE/NORAD has received a total of $27 million, and its units have a weighted average availability of 65 percent. Use of *F3* to allocate funds overrides any previous allocation.

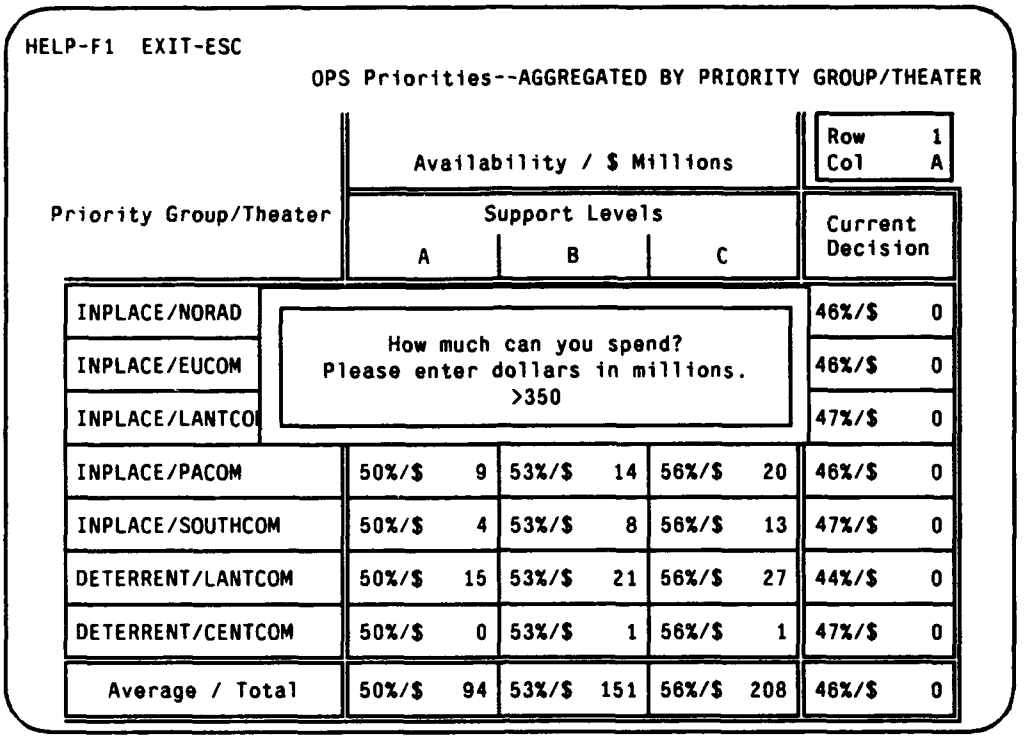


FIG. 5-6. TOTAL FUNDING MENU *(F3)*

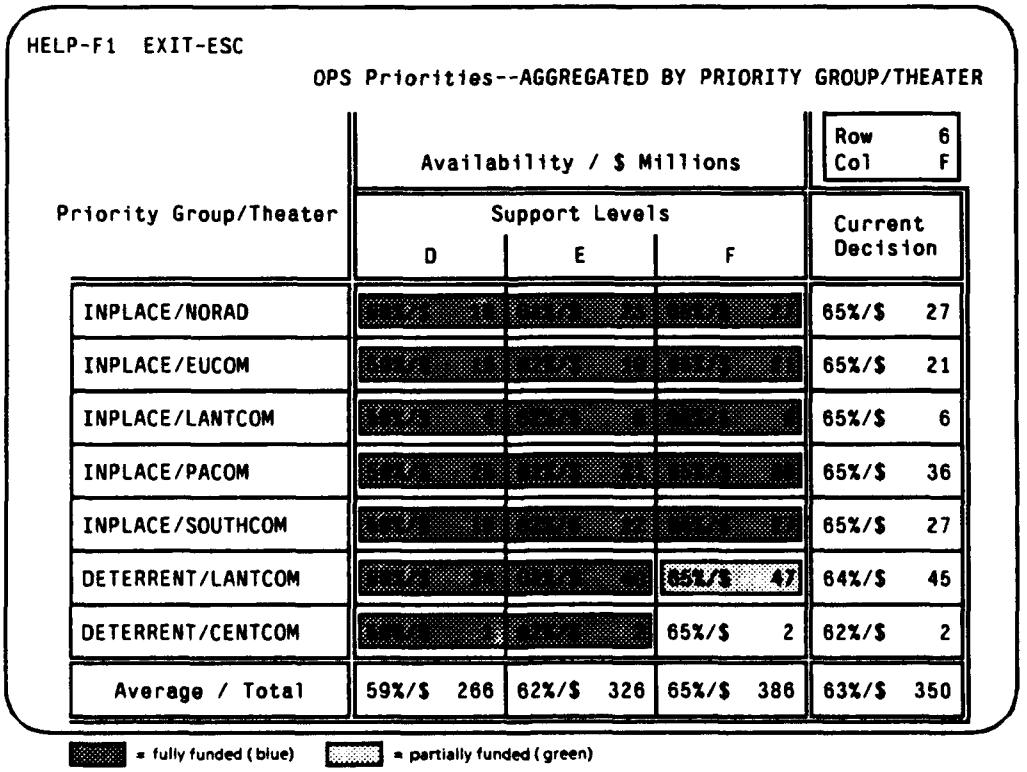


FIG. 5-7. DISPLAY AFTER SETTING TOTAL FUNDING TO $350 MILLION

PERSPECTIVES ON THE FUNDING DECISION

To view the results of a funding decision from another perspective, press *F2*, which causes the aggregation menu to appear (see Figure 5-8). The choice of aggregation is highlighted in red, with units aggregated by priority group/theater being the default. Use the up and down arrow keys to select the desired aggregation. Once you have made your choice, press *Enter*. In Figure 5-9, we show the display after selecting MD. The total expenditure is still $350 million and the weighted average availability is still 63 percent, as it was in Figure 5-7. From the second row, we can see that the result of this decision is that A-10 aircraft have a weighted average availability of 65 percent, and we have spent $22 million on kits for them.

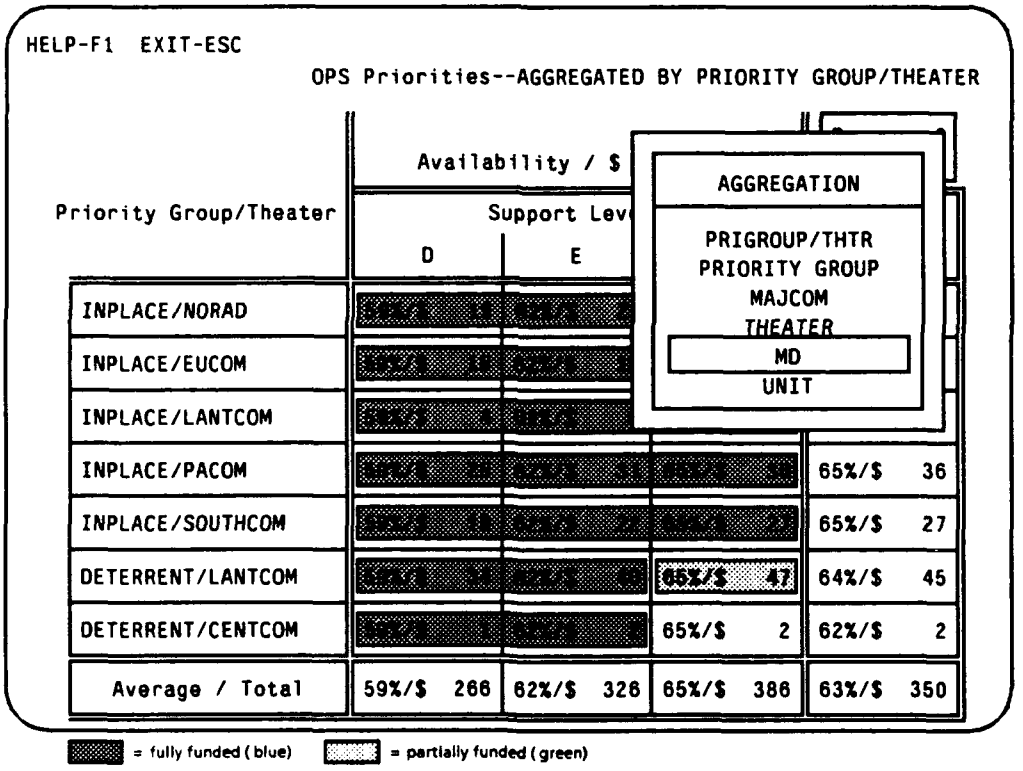


FIG. 5-8. AGGREGATION MENU *(F2)*

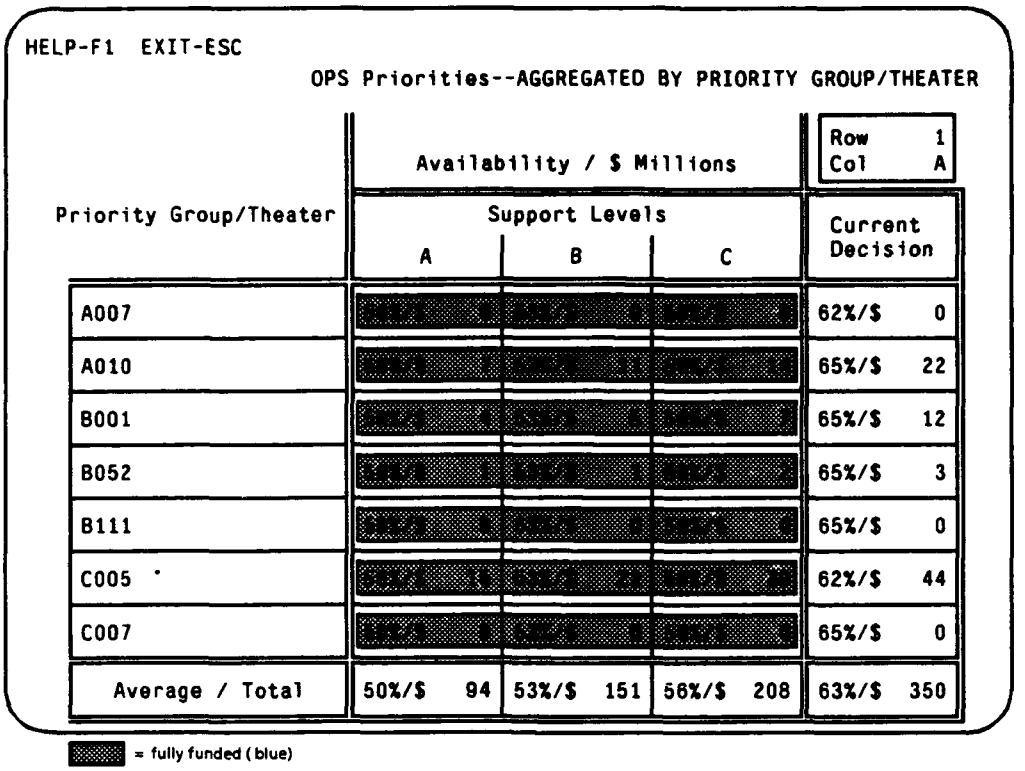


FIG. 5-9. DISPLAY AFTER SELECTING MD FROM AGGREGATION MENU

If we are in the MD aggregation, pressing the *+ key* will cause funds to be added to MDs in the order in which they are listed as row labels. That approach is not desirable unless the MDs have been ordered by priority. In our example, MDs are arranged in alphabetical order, so we would not wish to perform such an allocation. Use of the *+* or *– keys* should be limited to cases in which you wish to change the funding according to the operational priorities matrix method, funding one column at a time, from top to bottom.

Important: Use of the *+* or *– keys* in a new aggregation will completely override the old decision. If the aggregation is MD, existing funds (plus or minus the incremental funding) will be allocated to the matrix by bringing each MD to Support Level A, then B, then C, and so on, until funds are exhausted. In our example, pressing *+* once would reset funding to zero, add $1 million to the $350 million in funds to be allocated, and then allocate the new total of $351 million among the MDs starting at the top of Column A. This allocation would override our earlier one.

However, you can add or remove funds from a particular row (in the current aggregation, an MD) with the *Ins* and *Del keys*. For example, suppose we wish to add $5 million to A-10 kits. We use the arrow keys to move the current cell to the row labeled *A010*. Assuming that we have not changed the funding increment size with the *F4 key*, we press the *Ins key* five times to add the $5 million to A-10 kits. Note that this increases total funding by $5 million (see Figure 5-10); if we wish to keep total funding constant, we need to remove $5 million from another MD. Suppose we choose the C-5 aircraft; we move the current cell to the appropriate row and press *Del* five times. Now total funding is again $350 million (see Figure 5-11). To return to the priority group/theater aggregation and view the consequences of our decision, we simply press the *F2 key* and select a priority group/theater.

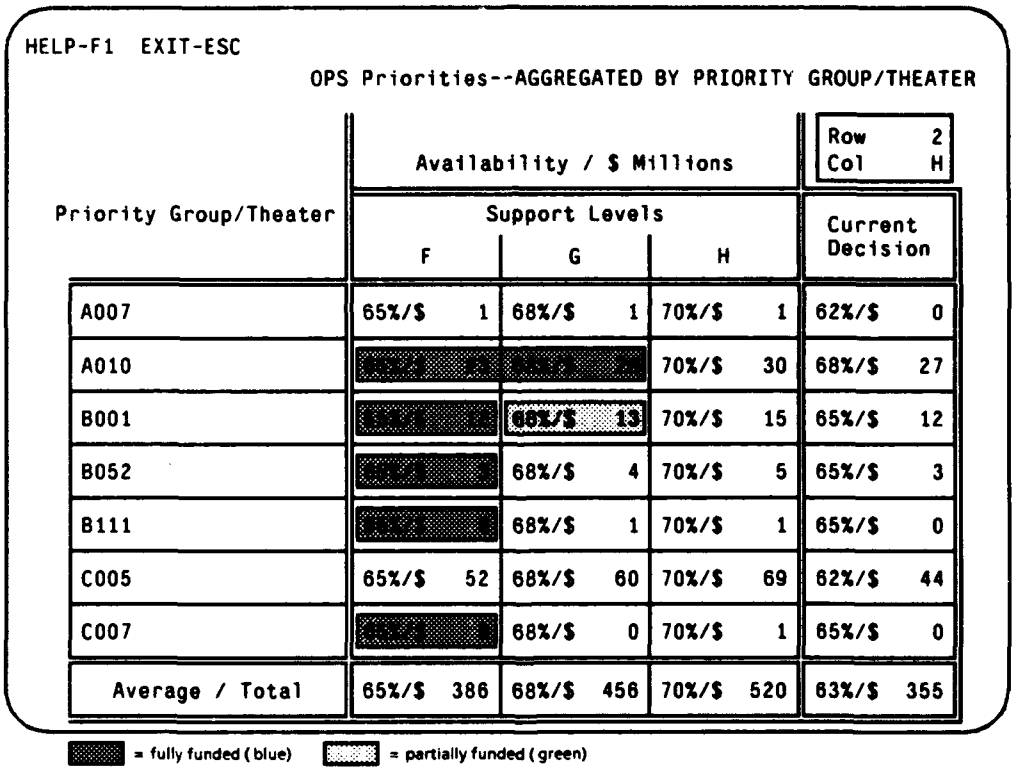


FIG. 5-10. DISPLAY AFTER ADDING $5 MILLION TO THE A-10 KITS *(INS)*

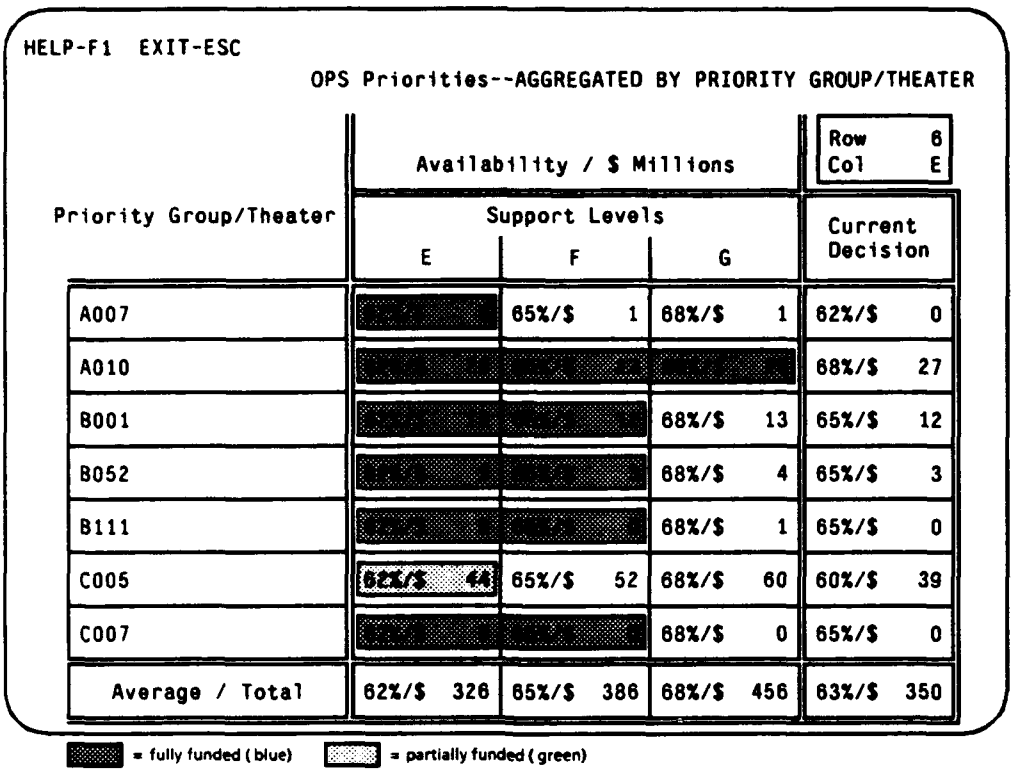


FIG. 5-11. DISPLAY AFTER REMOVING $5 MILLION FROM THE C-5 KITS *(DEL)*

Further discussion on allocating funds is presented in Chapter 6.

SAVING YOUR DECISION

To save the results of an OPSPRI session, press the *F5 key*. You will see a message informing you that the data are being saved to the appropriate disk files, as shown in Figure 5-12. This only saves your current decision for use later by the OPSPRI model. To save your decision for input to WSMIS, see the section on "Generating A WSMIS Output File" subsequently in this chapter.

GENERATING OUTPUT REPORTS

To produce an output report on a single row (a single unit, MD, priority group, etc.), first move the current cell to the row for which you wish to see a report. The column does not matter.

Press *F6*, bringing up the Report Generation Menu, shown in Figure 5-13. Use the arrow keys to select *CURRENT CELL ROW ONLY*, and press *Enter*.

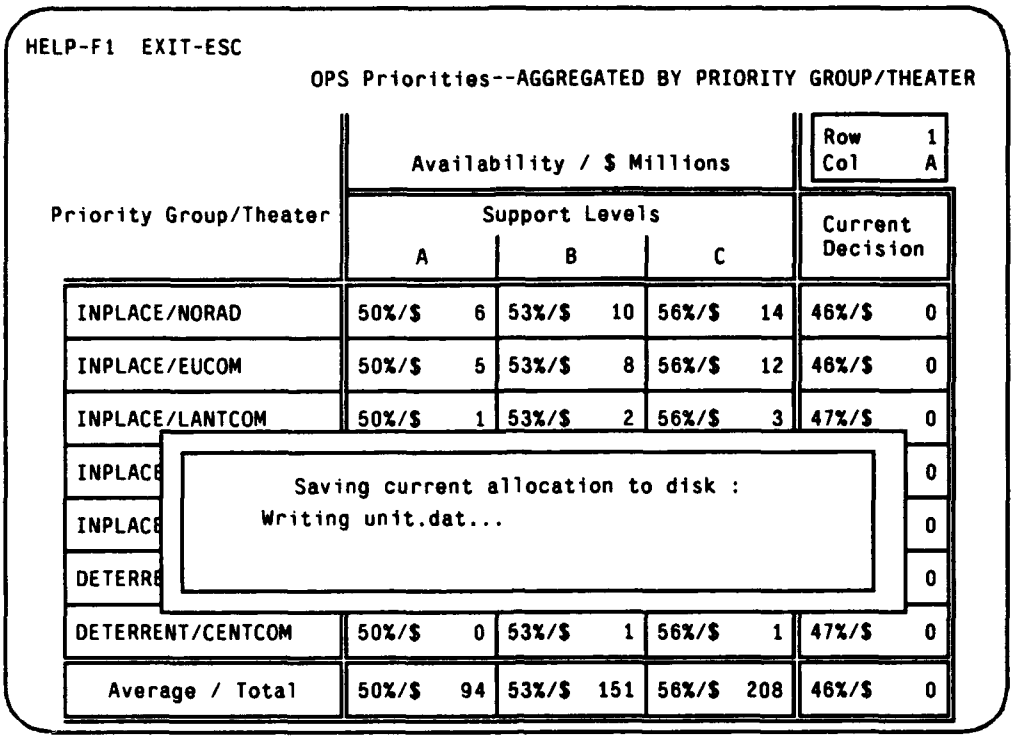


FIG. 5-12. SAVING DECISION

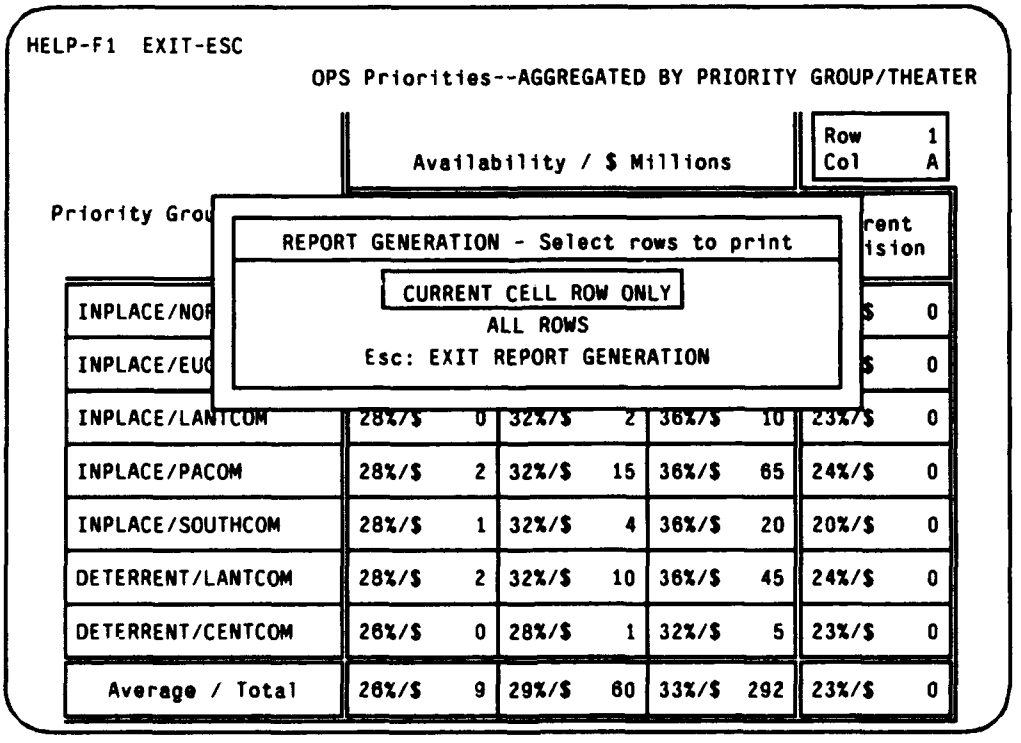


FIG. 5-13. REPORT GENERATION MENU

You will now see the Single Report Menu, shown in Figure 5-14. Use the arrow keys to select one of the four report types shown. Press *Enter* to confirm your selection. You may also press the *Esc key* to return to the previous menu.

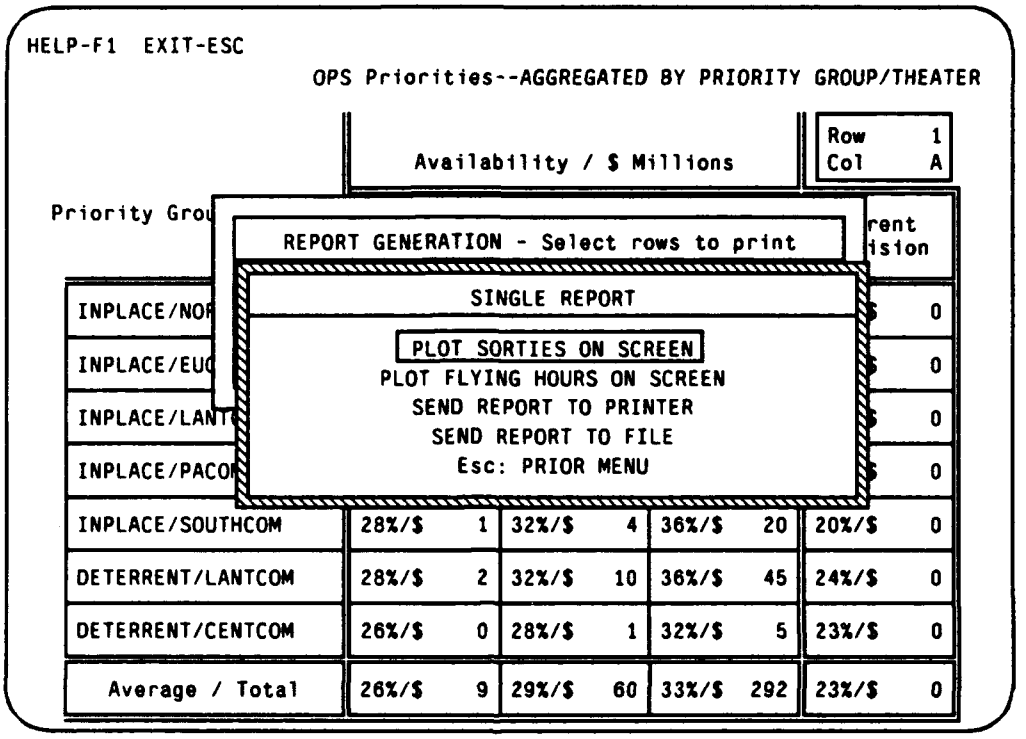


FIG. 5-14. SINGLE REPORT MENU

If you choose *PLOT SORTIES ON SCREEN*, you will see the message shown in Figure 5-15, followed by a graph similar to the one in Figure 5-16. Scheduled sorties by day are plotted in green; available sorties are shown in red. Press Esc to return to the Single Report Menu. *PLOT FLYING HOURS ON SCREEN* shows flying hours by day instead of sorties.

If you select *SEND REPORT TO PRINTER*, you will see the display in Figure 5-17. Your printer should then begin printing; if it does not, check to see whether it is on-line, and that all the connectors are in place. Use *Ctrl-Break* to leave OPSPRI if the printer still does not respond.

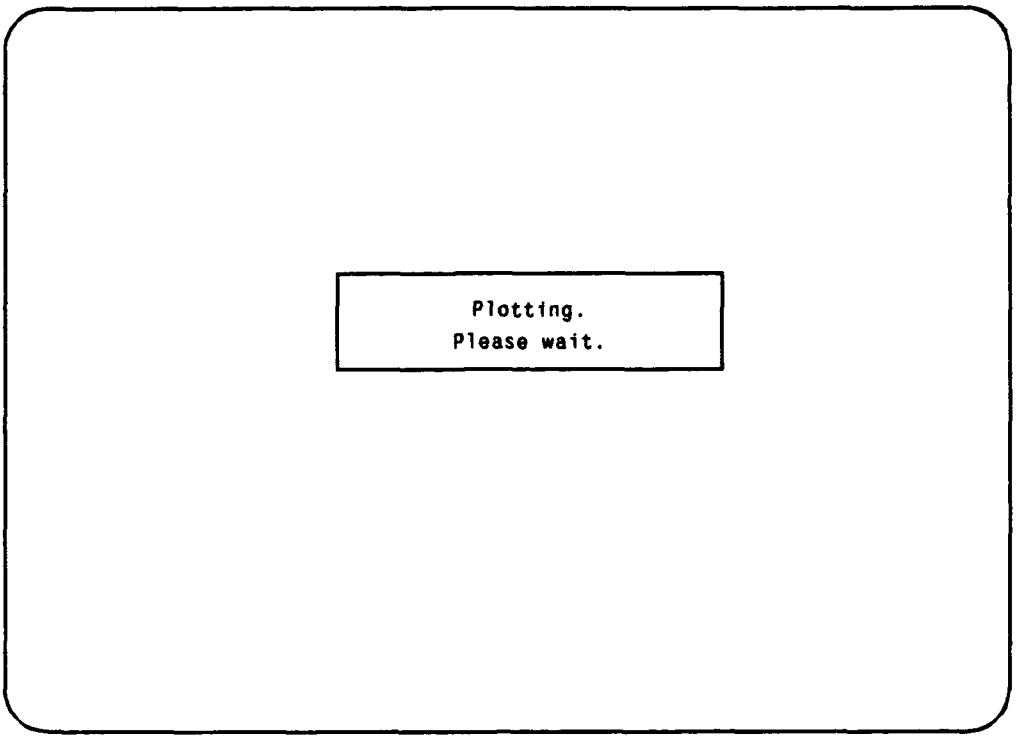


FIG. 5-15. SCREEN AFTER SELECTING PLOT

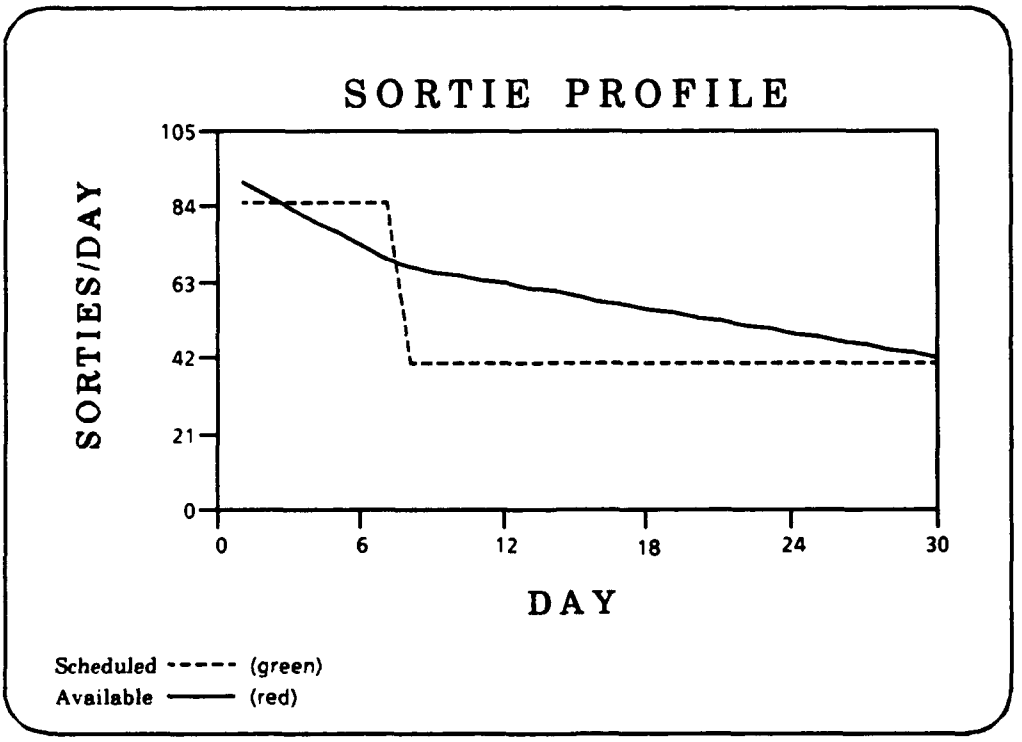


FIG. 5-16. PLOT OF SORTIES/DAY

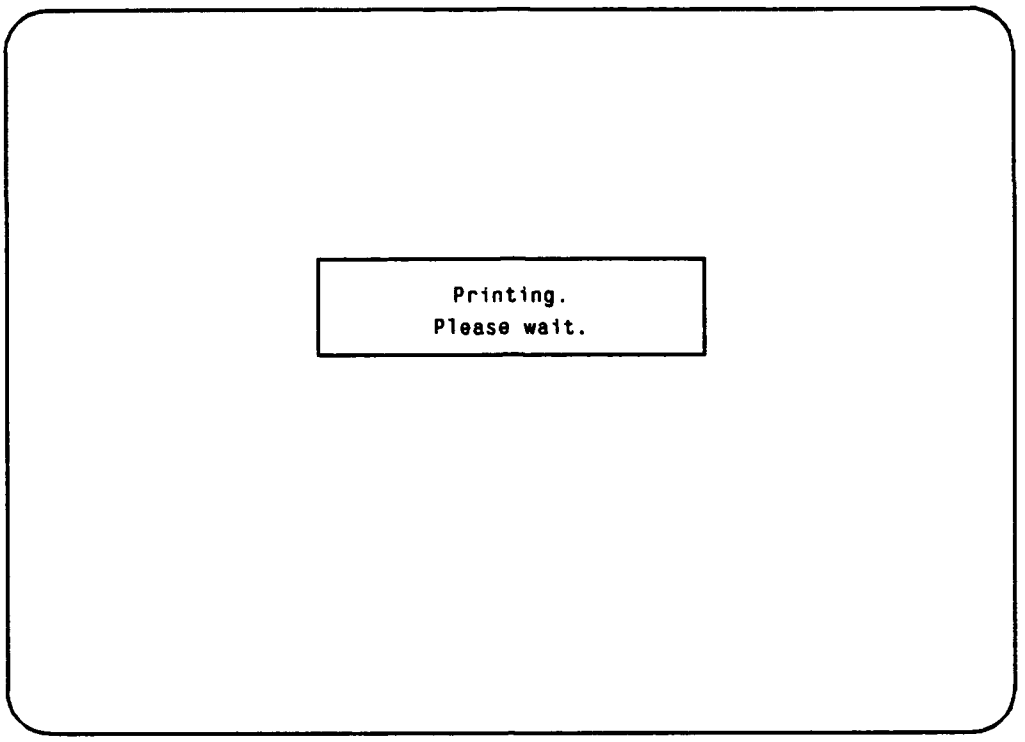


FIG. 5-17. SCREEN AFTER SELECTING PRINT

You will see a message *Done Printing* on the Single Report Menu when OPSPRI has finished sending data to the printer. Printing may continue, but you are now free to use the model.

When you choose *SEND REPORT TO FILE*, you will see the File Name Menu, shown in Figure 5-18. The report will contain both sorties and flying hours by day for the current cell row.

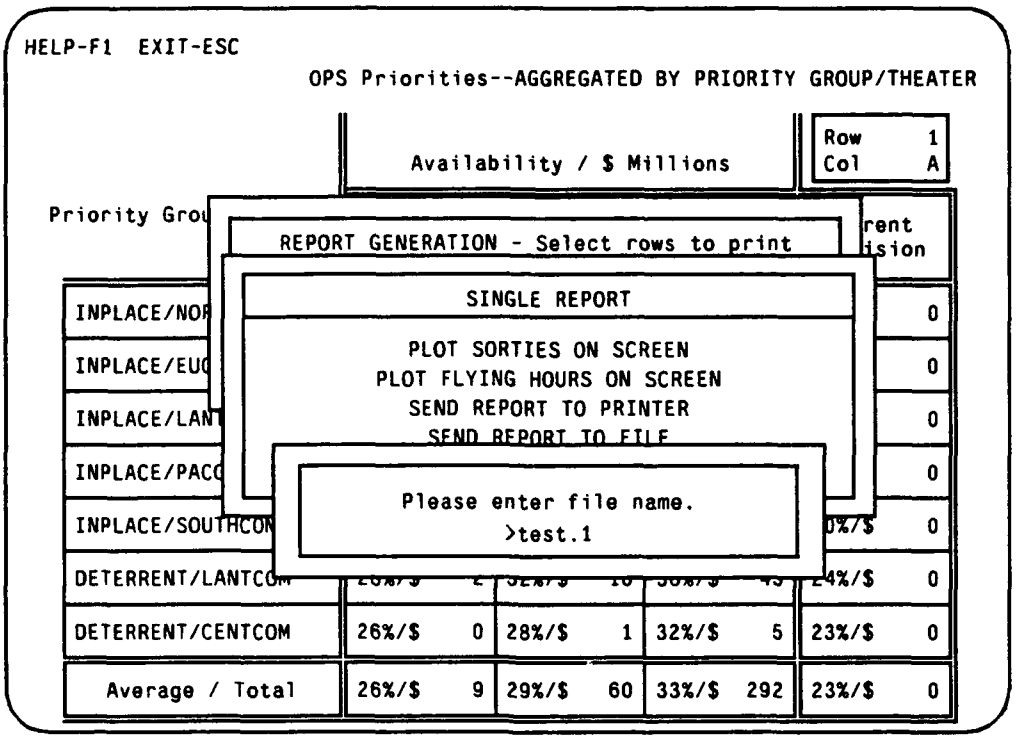


FIG. 5-18. FILE NAME MENU

Enter a valid DOS file name of no more than 12 characters, including the period. (In Figure 5-18, we have chosen the file name TEST.1.) Do not use file names with the following extensions; these extensions are reserved for the OPSPRI program and data files:

- .CR1
- .CR2
- .CRV
- .DAT
- .SCN
- .PAS
- .EXE
- .TPU
- .BGI
- .CHR.

If the file name you have selected is already in use, you will see the Overwrite Permission Menu, shown in Figure 5-19.

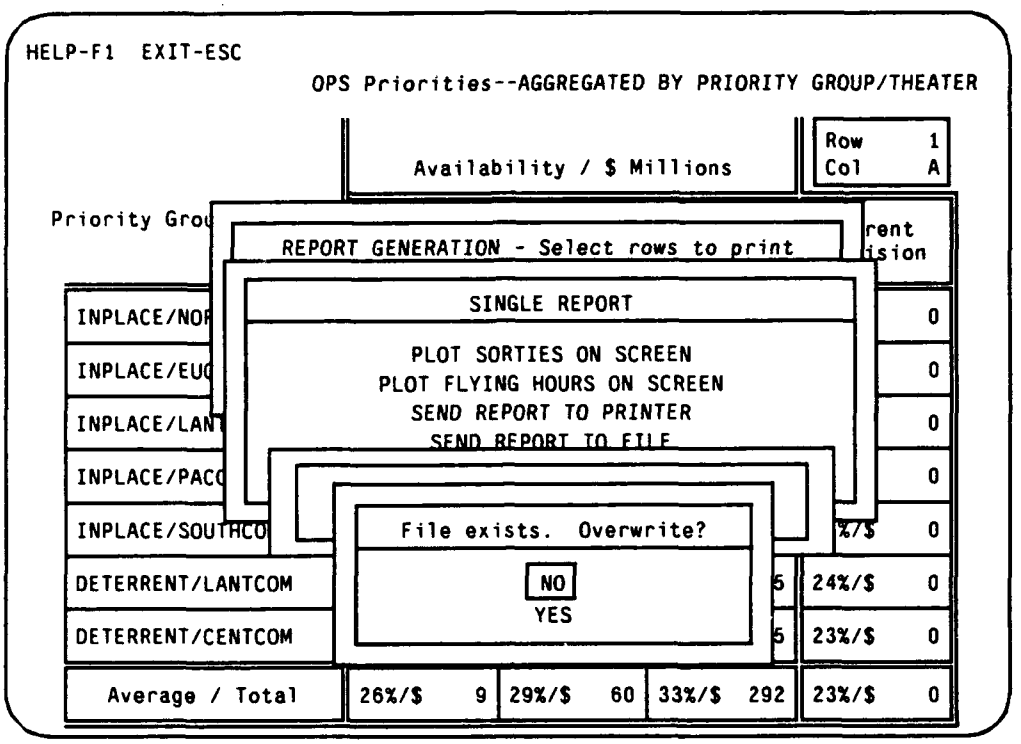


FIG. 5-19. OVERWRITE PERMISSION MENU

If you do not wish to overwrite the file, press *Enter* or type "N" to confirm *NO*. You will again see the File Name Menu, as shown in Figure 5-20; enter a new file name.

If you wish to overwrite the file, use the arrow keys, followed by *Enter* to select *YES*. You may also enter "Y."

If you choose a reserved file name, you will not be allowed to write to the file. Instead, you will see the reserved file name display, shown in Figure 5-21. Press *Esc* to bring up the File Name Menu, as shown in Figure 5-22, and enter a valid file name.

You will now see the message shown in Figure 5-23, followed by the Single Report Menu, showing the message *File Written*.

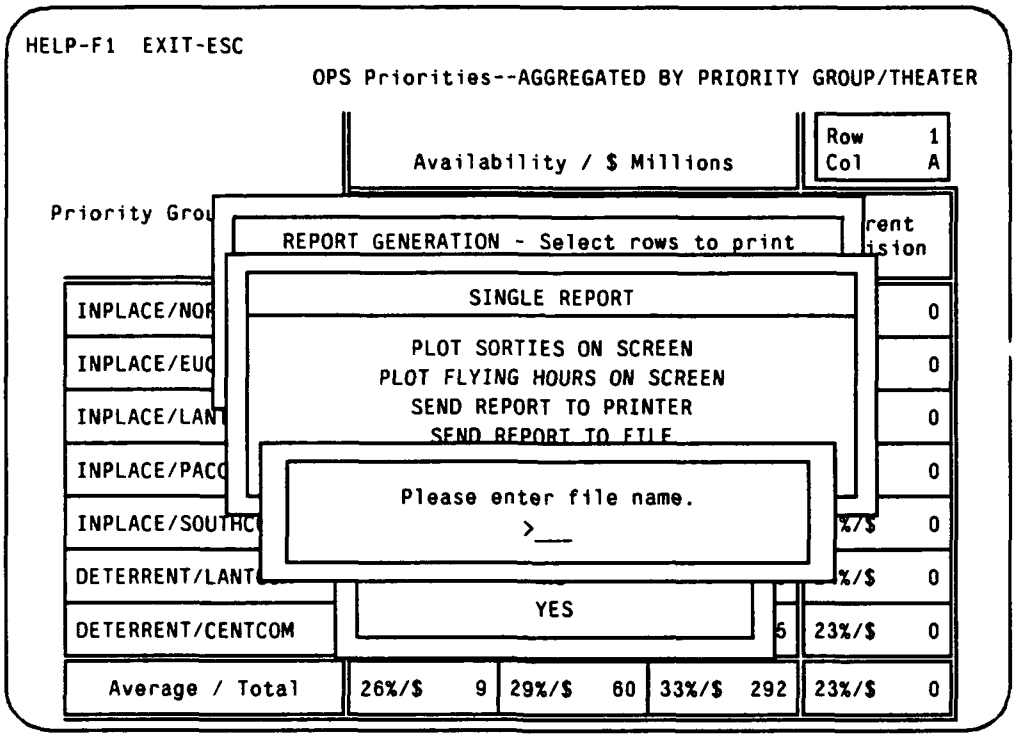


FIG. 5-20. SCREEN AFTER DECIDING NOT TO OVERWRITE EXISTING FILE

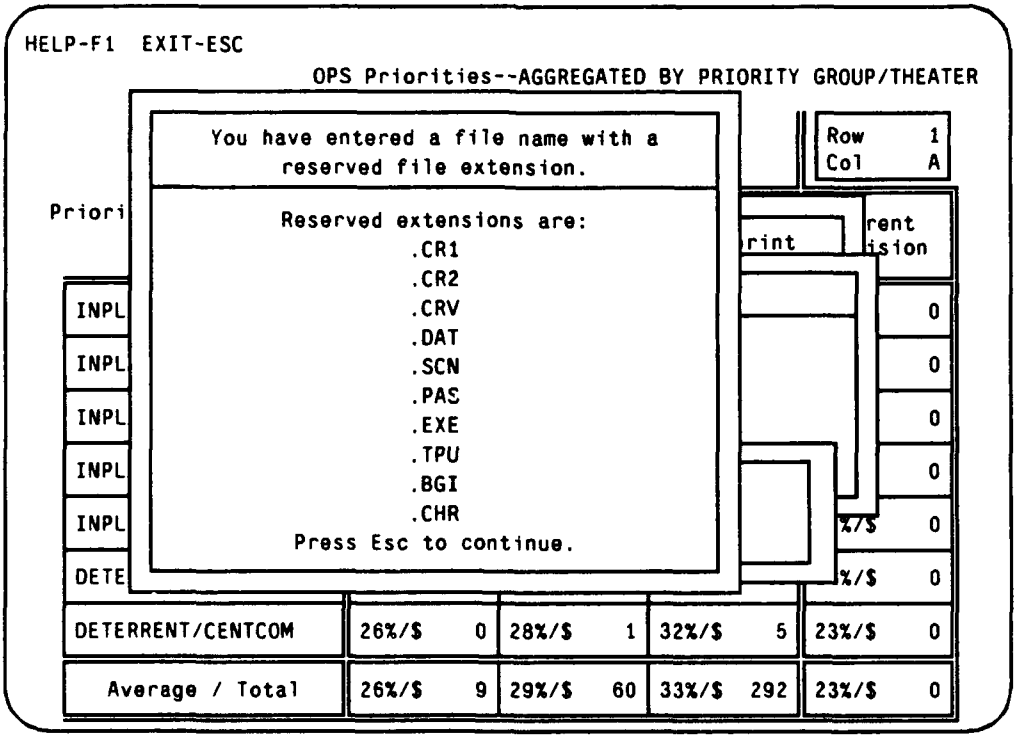


FIG. 5-21. RESERVED FILE NAME DISPLAY

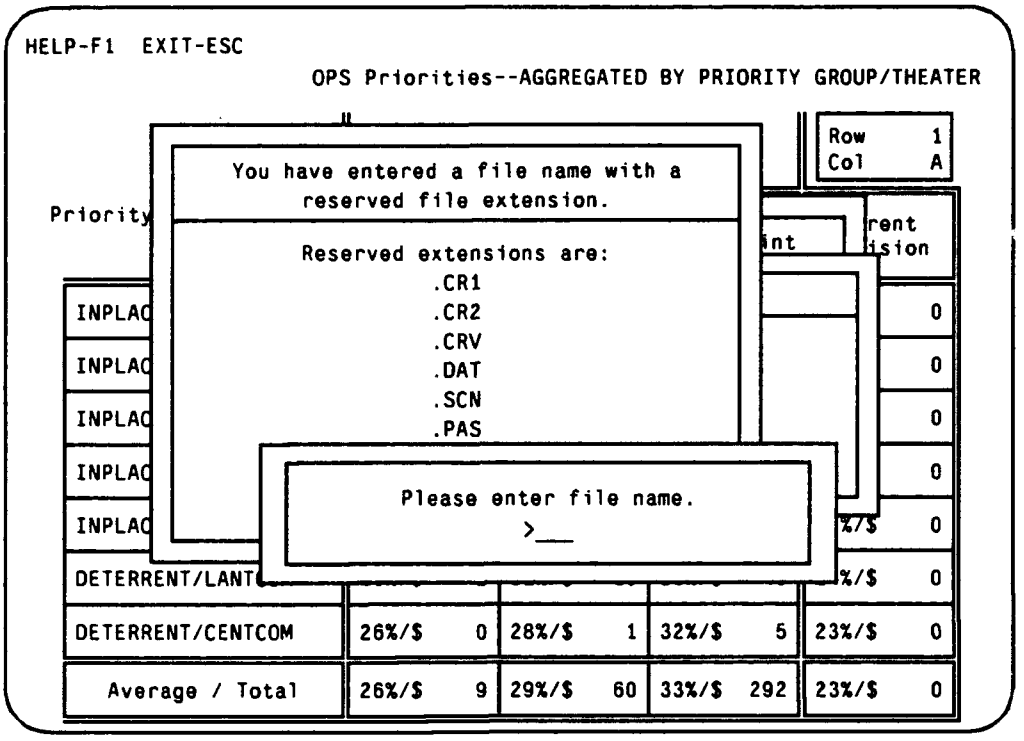


FIG. 5-22. SCREEN AFTER PRESSING *ESCAPE* ON RESERVED FILE NAME DISPLAY

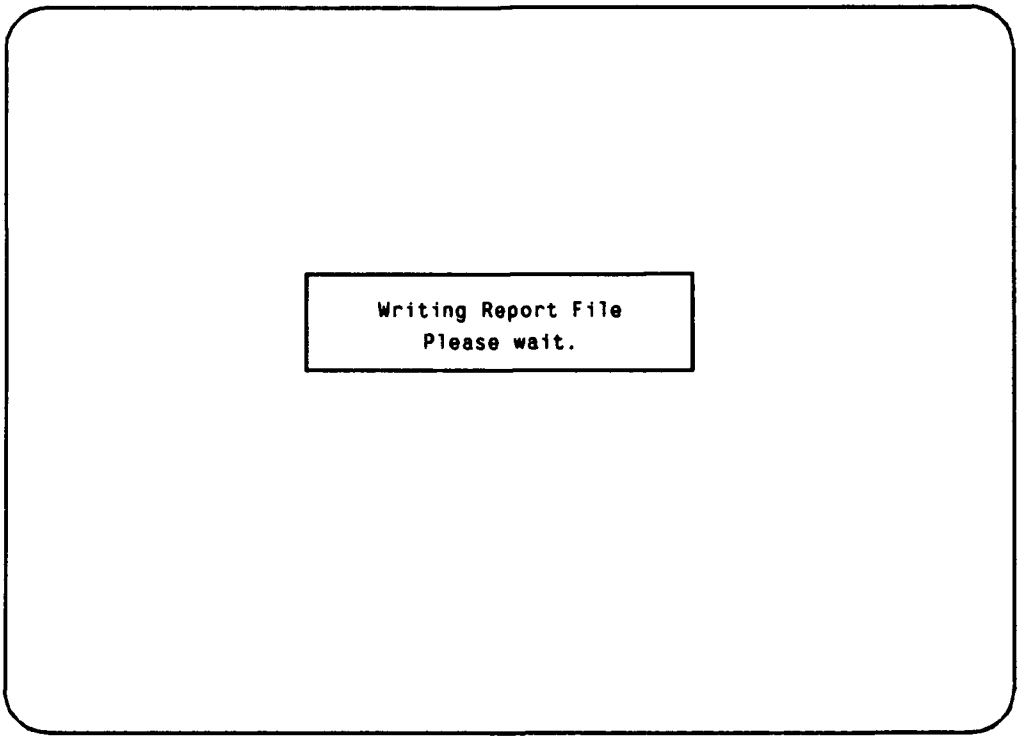


FIG. 5-23. SCREEN AFTER FILE NAME IS ACCEPTED

If you select *ALL ROWS* on the Report Generation Menu, you will see the Full Report Menu, shown in Figure 5-24. You may decide to send a report either to the printer or a file; in both cases, the report will contain both scheduled and available flying hours and sorties by day. The procedure for printing or writing to a file is the same as described previously for the current cell row.

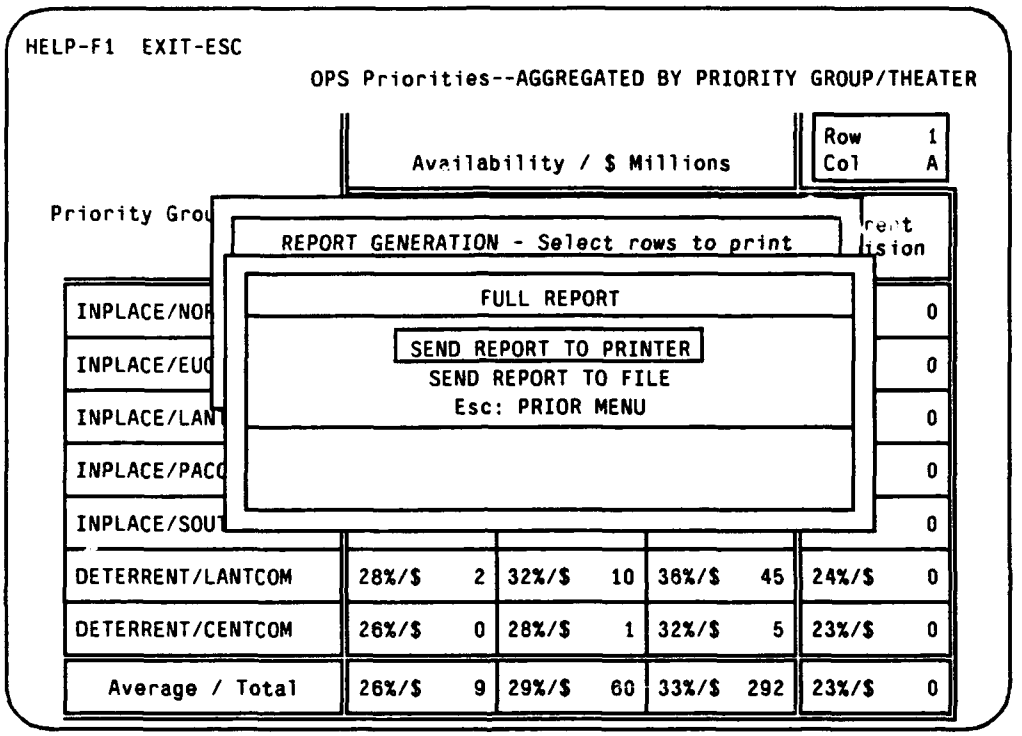


FIG. 5-24. FULL REPORT MENU

GENERATING A WSMIS OUTPUT FILE

To save the results of a funding decision to the output file UNIT.OUT, press *F7*. You will see a message telling you that the results are being saved. The file UNIT.OUT is a text file that shows each unit, together with its allocated dollars and its resulting availability. That data file is the one needed by WSMIS to perform budget allocation.

If you are in an aggregation other than the unit level, another file, showing the current decision cost and availability for each row, is also written. Output file names are as follows:

- *UNIT.OUT* – unit level
- *MD.OUT* – MD level
- *CMND.OUT* – MAJCOM level
- *THTR.OUT* – theater level
- *PRGP.OUT* – priority group level
- *PRGPTHTR.OUT* – priority group/theater level.

Output file format is shown in Appendix B.

OBTAINING HELP

As long as the operational priorities matrix display is visible and no menus are visible, you can get help by pressing *F1*. Press *Esc* to exit the help screen.

LEAVING THE MODEL

To leave OPSPRI, press *Esc*. If you have saved your decision, your funding allocation will be as you left it when you start the next session (provided you select *NO*, do not rewrite the data base, when you start the next session).

CHAPTER 6

MODELING FOR POLICY GUIDANCE

The OPSPRI model may be used to allocate WRSK/BLSS funds in one of two distinct ways – by priority order or by row. The model then enables the user to observe the consequences of the funding decision from a variety of perspectives. The user can also examine the sensitivity of aircraft availability (or cost) to input variables.

ALLOCATING FUNDS IN PRIORITY ORDER

In the priority order method of allocation developed by the Ops/Log Working Group, funds are first spent to bring all priority group/theater combinations up to Support Level A, starting with the highest priority combination and working downward. The model then repeats the process for Support Levels B, C, and so on until funds are exhausted. This method is used when the decision maker uses the +, −, or *F3 keys*, as discussed in Chapter 5. We refer to this method as "waterfill" because it is analogous to the process of filling one barrel with water (bringing all units to Support Level A), then a second (bringing all units to Support Level B), and so on.

Waterfill is a way of allocating funds to priority groups/theaters, but it can also be applied to funding by unit or by any aggregation of units. As long as the rows of the matrix are in priority order, where being higher in the matrix means having a higher priority, the waterfill method can be used. For example, if units are listed in order of priority, waterfill can be used to allocate funds at the unit level. If, however, MDs are in alphabetical order in the matrix, it does not make sense to use waterfill since an MD's location in the matrix does not reflect its priority.

If waterfill is used at any aggregation above the unit level, units receive funding as follows: For example, consider the units belonging to some Row, R. First, if Row R is fully funded through Support Level H and no more funds for Row R remain, all units belonging to Row R are fully funded through Support Level H and are not funded beyond that level.

If Row R is fully funded through Support Level G and partially funded through Level H, it has received some fraction of the funds it needs to bring it from Level G to Level H. Each unit belonging to Row R receives funds to support it at Level G plus that same fraction of the funds it needs to go from Level G to Level H. Each unit's availability is Level G plus that fraction of the increase in availability obtained in going from Level G to Level H.

To illustrate the second case (the one in which some levels are partially funded), suppose that the aggregation of data is priority group/theater, that the waterfill method is used to allocate funds, and that funds run out with INPLACE/EUCOM partially funded through Level H. Suppose that it would take $50 million to bring INPLACE/EUCOM from Level G to Level H, but that only $30 million is available to move it there. Then the fraction of funds needed to go from Level G to Level H is $f = 30/50 = 0.6$. Suppose the (fictional) 1000th TFS belongs to INPLACE/EUCOM, its Level G is 60 percent availability, its Level H is 65 percent availability, and the cost of going from Level G to Level H is $2 million. Then the 1000th TFS would receive enough funds to bring it to 60 percent availability plus $0.6 \times \$2{,}000{,}000 = \$1.2$ million, and would have a relative availability of $60 + [0.6(65 - 60)] = 63$ percent.

If the waterfill method is used to allocate funds at the unit level, all aggregations of units, such as F-15 or INPLACE/EUCOM, simply receive the sum of the funds allocated to their constituent units and are assigned the weighted average availability of those units, where each unit's weight is its PAA.

It is important to remember that if the waterfill method is used to allocate funds at one level of aggregation and then at another level, the results of the first allocation will be overridden. That is, total funding will first be reset to zero, and then new total funding will be allocated. For example, if we use the waterfill method to allocate $350 million to theaters and then change our perspective to the unit level and allocate $350 million again, the first $350 million allocated will be removed, leaving zero funds, and then the new $350 million will be allocated among units using the waterfill method.

The order in which a unit is funded by the waterfill method is determined by its position in the file WMP3.DAT (Chapter 3 presents a discussion of this file). For theaters, priority groups, MAJCOMs, and MDs, the priority order is determined by

the order in which the entity is listed in the list files (again, Chapter 3 presents a description of list files).

ALLOCATING FUNDS BY ROW

In any aggregation, funds may be allocated directly to, or removed from, a row, using the *Ins* and *Del keys*. For example, suppose we are working at the MD level and wish to delete $10 million from A-7 kits. We position the current cell on the A-7 row and use *Del* to remove the funds. When we do this, the A-7 will either be fully funded through some support level, with no funds remaining; fully funded through a support level and partially funded through the next support level; or not funded. In the first case, all A-7 units would be fully funded through the same support level. In the second case, the A-7 has received some fraction, f, of the funds it needs to go from its last fully funded support level to the next level. Each unit receives funds to take it to that same fully funded support level plus the fraction, f, of the funds it needs to go from that last fully funded support level to the next level.

If we are working at the unit level, we may add or remove funds one unit at a time. That feature would be useful, for instance, if a new unit were introduced. That unit's MAJCOM, MD, priority group, and theater will have its current cost and resulting weighted average availability recomputed to reflect the change.

SENSITIVITY TO CHANGES IN KIT REQUIREMENTS

If WRSK/BLSS requirements change, OPSPRI can show the effects of that change on either availability relative to requirements (holding funds constant) or on funds needed to maintain a fixed availability relative to the requirements. Kit requirements may change in response to changes in the following factors:

- The DSO either for the initial surge or for the subsequent sustainment part of the scenario
- The flying program
- Failure rates for kit components
- WSMIS/REALM's method for computing aircraft availability curves.

Changes in WRSK/BLSS requirements change the corresponding aircraft availability curve files and may change the scenario files as well. No other input file types need to be changed.

If the total funds are constant, you can use the OPSPRI model to compare the new relative availability for each priority group/theater, priority group, theater, MAJCOM, MD, and unit to the old one. You can also use the model to compare the new and old estimated flying hours and sorties. First, using the old data, make output files showing current decision costs and availabilities, as well as reports showing sorties and flying hours. Copy the old curve files to another directory and replace them with new ones. Replace the scenario files as well if they have changed. Run OPSPRI, re-create the data base, enter the same total funding, and follow the same steps you used before. Then look at the new availabilities in the current decision cells and/or the new sorties and flying hours in the output report and compare them with the old values.

For example, suppose that we allocated a total of $350 million to priority groups/theaters using the waterfill method and then we went to the MD level and removed $10 million from the A-7, transferring it to the A-10. With the new curve files (and scenario files, if changed), we would first re-create the data base, then allocate the $350 million as before, this time producing different availabilities for the priority groups/theaters. We would again go to the MD level and transfer $10 million from the A-7 to the A-10. The resulting availabilities for MDs, priority groups/theaters, etc., could then be compared with those obtained with the old curves.

If you wish to hold the relative availability constant, save the old results as above and using new data, bring each group of forces (priority group, unit, etc.) to the same relative availability it had before. The total cost of keeping relative availability the same with the new requirement is shown in the Current Decision cell at the bottom right. The cost for each unit, priority group, etc., is shown in the Current Decision cell at the end of its row. Sorties and flying hours can be compared as above, where cost was being held constant.

For example, suppose that with the old requirement (old availability curves and scenario files), you had first allocated enough funds to bring all priority groups/theaters above INPLACE/LANTCOM in the matrix to Support Level H, INPLACE/LANTCOM to a relative availability between Level G and Level H, and all priority groups/theaters below INPLACE/LANTCOM to Level G. Suppose that you had then gone to the MD level, brought the relative availability for the F-16 up to 90 percent, and that for the A-10 up to 80 percent.

With the new curves and scenario files, you would again bring each priority group/theater above INPLACE/LANTCOM to Level H, INPLACE/LANTCOM to the same relative availability it had before, and all priority groups/theaters below INPLACE/LANTCOM to Level G. (Note that while the cost of reaching these support levels will be different from the old cost, the relative availability will remain the same.) You would again go down to the MD level and bring the F-16 and A-10 to 90 and 80 percent availability, respectively. The new total cost can be read from the bottom right-hand corner cell, and the new cost for a particular MAJCOM, theater, etc., can be read from the current decision cell at the end of its row.

SENSITIVITY TO CHANGES IN FORCE STRUCTURE

Some changes in force structure, bed-down, or priority that can change the results of WRSK/BLSS funds allocation are:

- Changing the priority group, theater, or MAJCOM assignment for units
- Adding or deleting units
- Changing the PAA of existing units.

Before-and-after comparisons may be made either keeping total dollars constant or keeping the relative availability of priority groups/theaters, MDs, or units constant. To do so, first, print output reports containing flying hours and sorties and make output files containing costs and availabilities.

If the priority group, theater, MAJCOM, or PAA for a unit has changed, edit the file PRIORITY.DAT to reflect that change; if the unit belongs to a new theater, add that theater to THTRLIST.DAT; if the unit belongs to a new priority group, add that priority group to PRGPLIST.DAT; if the unit belongs to a new command, add that command to CMNDLIST.DAT; and if the unit employs a new MD, add that MD to MDLIST.DAT.

If a unit has been deleted, delete that unit from PRIORITY.DAT.

If a new unit has been added, add a record (line) to PRIORITY.DAT to reflect the addition; verify that there is a REALM curve file (or files) and a scenario file for this unit; edit CROSSREF.DAT so that the unit and its curve file names and scenario file name appear. A record must be added to CRVFILES.DAT showing the path/file names of the availability/cost curves for this unit, and MAKECRVS must be rerun to

create the relative availability curve file for the new unit. If the unit belongs to a new priority group, theater, or command or if it uses a new MD, add the new item to the appropriate list file (for example, MDLIST.DAT for a new MD).

Now allocate funds as before, either keeping cost or availabilities constant, and print the new output files and reports. Illustrations of this process appear in the previous section.

SENSITIVITY TO CHANGES IN PRIORITIES

Any change in the relative priorities of theaters, MAJCOMs, MDs, or units can change WRSK/BLSS allocations. If such a change occurs, save the output reports and files containing availabilities, costs, flying hours, and sorties. Edit the appropriate list file so that the items appear in order of their priorities, where higher priority means "nearer to the top of the list." Now run OPSPRI again, allocate funds as before (examples can be found in this chapter under the "Sensitivity to Changes in Kit Requirements" section) and print the new output reports/files.

CHAPTER 7

INSIDE OPSPRI

We wrote, compiled, and linked OPSPRI's source code using Borland's Turbo Pascal 5.5. (A description of hardware and software needed to run the model is given in Chapter 2.) The remainder of this chapter describes model inputs, disk data base creation, screen data base creation, screen manipulation, saving decisions, and generating output reports. The description is simplified for brevity, but it provides a general idea of how OPSPRI works.

INPUTS

The model utilizes the following input files:

- *OPSPRI.EXE* – contains the OPSPRI program
- *Relative availability curve files* – one for each unit
- *Scenario files* – one for each unit
- *CROSSREF.DAT* – provides curve and scenario file names for each unit
- *PRIORITY.DAT* – contains MD, MAJCOM, theater, priority group, and support levels for each unit
- *MDLIST.DAT* – ordered list of MDs
- *CMNDLIST.DAT* – ordered list of MAJCOMs
- *PRGPLIST.DAT* – ordered list of priority groups
- *THTRLIST.DAT* – ordered list of theaters
- *Graphics files:*
  - ATT.BGI
  - CGA.BGI
  - EGAVGA.BGI
  - HERC.BGI

- IBM8514.BGI
- PC3270.BGI
- SANS.CHR
- TRIP.CHR.

These files are described in greater detail in Chapter 3, input file formats are listed in Appendix A.

DISK DATA BASE

The OPSPRI model first determines whether to create the disk data base, which is used to store data for the next session while the model is not in use. The model searches the hard disk for data base files. If one or more of those files is missing, it creates the disk data base from the input files listed above. Otherwise, it prompts the user to choose between using the existing data base or re-creating it, using new input files.

The disk data base is created by processing one unit at a time. The model reads a record of the file PRIORITY.DAT, obtaining a unit and that unit's priority group, theater, MAJCOM, MD, sortie length, and set of support levels. Next, it searches CROSSREF.DAT and finds the unit's relative availability curve and scenario files. For each support level, it interpolates the relative availability curve to estimate the cost of attaining that support level. When that activity is completed, it writes a disk data base record for that unit, containing the unit name and a cost/availability pair for each support level. It also writes another record for the unit, containing curves for the primary and secondary days, the sortie length, and the scenario file name. (See the subsequent section in this chapter on "Generating Reports" for an explanation of primary and secondary days.) The curves are used later by the report generator to produce estimated sorties and flying hours. OPSPRI then goes back to PRIORITY.DAT and reads the next unit.

Temporary data structures are used to accumulate costs and availabilities for MDs, MAJCOMs, theaters, priority groups, priority groups/theaters, and the totals row of the display. After OPSPRI finds the cost to go with one of the support levels for a unit, it finds that unit's MD in the MD temporary data structure and adds the cost to the running total for the MD. That unit's contribution to the availability total for the support level is first weighted by the unit's PAA and then added in. That

procedure is followed for each support level. A similar process is followed for adding that unit's cost and availability contributions to the MAJCOM, theater, priority group, priority group/theater, and totals data structures.

For each MD, a temporary data structure records the units that contributed to the MD's cost and availability totals, keeping them in an ordered list. A running total for the MD's PAA is also maintained. Once all units are processed, availabilities for each MD are divided by their respective PAA's, producing weighted average availabilities, and MD records are written to disk. Each record contains the MD name, a cost and availability pair for each support level, and a reference to the list of units that belong to that MD. A similar process is followed for each MAJCOM, theater, priority group, priority group/theater, and the totals record.

Ordered lists of units belonging to MDs are written to one file, ordered lists of units belonging to MAJCOMs to another, and so on. Another file records the number of units, MDs, MAJCOMs, theaters, etc. These files are used to create the screen data bases.

SCREEN DATA BASES

The OPSPRI model owes its quick response to its use of memory – all data needed to allocate funds reside in RAM. Disk access is only required at the start of a session, when saving a decision, or when generating a report. While OPSPRI is running, data are stored in five screen data bases, each corresponding to an aggregation type: one for units, one for MDs, one for MAJCOMs, etc. The screen data bases are dynamic data structures that expand or contract to fit the number of units, MDs, theaters, and so on, reducing the amount of memory needed, as well as the time needed to locate data needed in computations.

Records in each screen data base correspond to rows of the matrix display. Each row record contains the row name, a cost/availability pair for each support level, a pair for the Current Decision column, and the list of units belonging to that row.

SCREEN MANIPULATION

The user may manipulate the screen display in four ways: scrolling, waterfill, funding by row, and reaggregation.

The user scrolls with the *arrow keys*, *PgUp* and *PgDn keys*, and a number of other keys (see Chapter 5). As these keys are pressed, the current cell coordinates change. If the new current cell location is outside the display window, OPSPRI moves the window. The new section of the matrix displayed extends just far enough to show the new current cell. For example, suppose that the current cell is in Row 1, Column 3, and Columns 1 through 3 are visible. If we press the *right arrow key*, the window must move to show Columns 2 through 4, with the current cell in Row 1, Column 4.

The waterfill method, as explained in Chapter 6, allocates funds by first bringing each unit up to a first level of support, starting at the top of a priority list and working down, then bringing each unit to a second level of support, and so on until the funds run out. The user does this by using the *F3*, *+*, and *–* *keys*. The current cell location is the last cell to receive at least partial funding, and as above, OPSPRI repaints the screen with data chosen from a section of the matrix containing the new current cell. Each time the *F3*, *+*, or *–* *key* is used, the current decision cost and availability in each row record of the screen data base are updated. The OPSPRI model also takes each row record in the current screen data base, finds the units associated with that row, and updates the current decision cost and availability for those units. Each of the row's units is fully funded through the same support level as the row record and receives the same fixed fraction of the funds needed to go from the last fully funded support level to the next one. The updated unit-level data base will be used to compute the Current Decision cell costs and availabilities of any new aggregation selected.

The *Ins* and *Del keys* are for funding by row. The OPSPRI model adds or subtracts dollars from the current cell's row, moves the current cell to the last funded cell on that row, moves the window to show that cell, and updates current decision cost and availability for both current and unit-level data bases as explained above.

The *F2 key* brings up the aggregation selection menu. When the user selects a new aggregation, the OPSPRI model takes each row record for the new screen data base, finds the units that belong to that row, and uses the current decision cost and availability for those units to compute the current decision cost and availability for the row record. The row record's current decision cost is set equal to the sum of the current decision costs of its constituent units; its current decision availability is set

equal to the weighted average of its units' availabilities, where each unit's weight is its PAA divided by the row's PAA.

SAVING A DECISION

When the user presses the *F5 key* to save the results of a decision, the OPSPRI model takes data from each screen data base and writes them to the corresponding disk data base file. If the user then leaves the model and returns later for another session, the most recent data saved will be used to create the screen data bases.

GENERATING REPORTS

When the user decides to plot, print, or write an output report to a file for the current cell row, OPSPRI first goes to the active screen data base and obtains the current funding and availability for each unit belonging to that row.

As a preliminary to the description of report generation, we briefly present the scenario assumptions for units and the way that those assumptions affect kit computations.

War plans for Tactical Air Force units are typically based on a two-part scenario. The first part consists of the first 7 days and requires a high sortie rate, and the second part consists of Days 8 through 30 and requires a lower sortie rate. Each part of the scenario has a DSO; it is the number of available aircraft required on the last day of that part of the scenario. REALM computes two aircraft availability curves for such units, one for Day 7 and one for Day 30. For some units, funds are first applied to the Day 7 curve. Once enough funds are allocated to meet the Day 7 DSO, remaining funds are allocated to increase the number of available aircraft on Day 30, using the assets purchased for Day 7 as initial assets. For others, funds are allocated to the Day 30 curve first. We refer to the day of the scenario whose curve receives funds first as the *primary* day and to the day whose curve receives funds second as the *secondary* day.

For strategic airlift units, the scenario involves only one sortie rate and one DSO, stated for Day 30. In this case, Day 30 is both the primary day and the secondary day.

From a unit's current funding and the cost of attaining each support level, the OPSPRI model determines the last fully funded support level. Next, it finds the

available fraction, f, of funds needed to go from the last fully funded support level to the next support level. The model reads this unit's record in the WMP.DAT data base. This record contains the sortie length, the primary day and its DSO, the secondary day and its DSO, and the availability curves for each of those days. Each curve has an availability/cost pair for each support level. OPSPRI determines which of these days is chronologically first (not first in funding order). We call the first of these days Day A and the second Day B.

Using the available fraction, f, of funds needed to go from the last fully funded support level to the next support level, together with the availability curve for Day A, OPSPRI interpolates availability for that day. It estimates Day B availability in a similar way.

Availability is converted to ENMCS. Assuming that all PAA were available on Day 0, OPSPRI interpolates ENMCS aircraft for each day between Day 0 and Day A. It then uses ENMCS on Day A and Day B to interpolate EMNCS for each day in between.

The model next reads the unit's scenario file, which contains the unit's scheduled flying hours for each day of the scenario. If scheduled flying hours are not provided for each day, it assumes that they remain constant from the last day for which they were provided until the end of the scenario (Day B). OPSPRI finds sortie length in the WMP.DAT record and computes scheduled sorties for each day as

$$(\textit{scheduled flying hours for day})/(\textit{sortie length}).$$

The OPSPRI model estimates scheduled flying hours per day for each aircraft on Day A and Day B, respectively, by

$$(\textit{scheduled flying hours on Day A})/(\textit{DSO for Day A})$$

and

$$(\textit{scheduled flying hours on Day B})/(\textit{DSO for Day B}).$$

Available flying hours per aircraft per day is set to the larger of these two rates. We call this the flying-hour rate.

The OPSPRI model then estimates the unit's available flying hours for each day by [(flying hour rate) times (PAA minus average of ENMCS for the day and the previous day)]. Dividing each of these numbers by sortie length yields estimated available sorties by day.

For each day of the scenario, the model sums scheduled flying hours, available flying hours, scheduled sorties, and available sorties across all units belonging to the row. Scheduled flying hours and estimated available flying hours by day are sent to the plotter, printer, or a file, depending on the user's choice. Sorties are treated in the same way. For plotting on the screen, the user must choose between flying hours or sorties; scheduled and available quantities are then plotted by day on the same graph.

If the user selects a report on all rows of the matrix, the above process is repeated until reports have been produced for each row. Reports for the rows are all sent to the same file or are all included in the same printer job depending on the output device selected. Plotting for all rows is not available.

PRODUCING OUTPUT FILES

When the user presses the *F7 key*, the model finds the unit-level screen data base, obtains the current funding and availability for each unit, and writes these data to the output file UNIT.OUT. If the active screen data base is not the unit-level data base, the OPSPRI model also writes current funding and availability for each row in the current screen data base to a separate output file. Chapter 5 presents a list of output file names used.

CHAPTER 8

TROUBLESHOOTING

This chapter describes error messages that can occur in attempting to run the OPSPRI model and the likely causes of those errors. While not comprehensive, the listing should help to resolve most problems. Error messages usually fall into one of three classes:

- *Matching Errors* – cannot find an item in the appropriate list file or cross-reference file.
- *File Opening Errors* – a needed input file is probably missing.
- *File Reading Errors* – a needed disk data base file is not in the correct format.

Examples of each type of error message, along with the probable cause and cure, are described below.

MATCHING ERRORS

If you see the message

> ERROR – Cannot match MD from unit 1000th TFS with any of the MDs on the MD list.

look at the row of the file PRIORITY.DAT containing the unit name 1000th TFS, and find its MD, listed in the same row. Compare that MD with the ones listed in the file MDLIST.DAT. You will probably find that either the MD for the 1000th TFS is not in MDLIST.DAT or the spelling or representation of that MD is not the same in the two files.

For example, the above error message would be produced if the MD for the 1000th TFS is F117, and F117 is not on the MD list. The same error message would also be produced if the MD for the 1000th TFS is listed as "F16" and the MD in MDLIST.DAT is listed as "F016". Names that appear the same but have a different number of spaces are not treated as equal. For example, " F016" is not the same as "F016".

Similar error messages, with MD replaced by MAJCOM, theater, or priority group, should be investigated by finding the unit name in PRIORITY.DAT and comparing its MAJCOM, theater, or priority group with those in the corresponding list files. The MAJCOM list file is CMNDLIST.DAT, the theater list file is THTRLIST.DAT, and the priority group list file is PRGPLIST.DAT.

If you see an error message

> Could not find unit identifier in cross-reference file that would match the unit 1000th TFS from the priorities file.

find the unit 1000th TFS in the file PRIORITY.DAT and compare it with the unit identifiers in the file CROSSREF.DAT. You will probably find that 1000th TFS is not on the cross-reference file list, or that it is represented differently in the two files. For example, the unit identifier could be "1000th TFS" in PRIORITY.DAT, but " 1000th TFS" in CROSSREF.DAT (*different numbers of embedded blanks*).

FILE OPENING ERRORS

If you see the message

> Cannot open input file PRIORITY.DAT. Please check that PRIORITY.DAT is in your input file directory.

the file PRIORITY.DAT is probably not in the same directory as OPSPRI.EXE (it must be) or your file name PRIORITY.DAT is misspelled.

A message that looks the same as the above message but that has the file name CROSSREF.DAT, MDLIST.DAT, PRGPLIST.DAT, THTRLIST.DAT, or CMNDLIST.DAT in place of PRIORITY.DAT should be diagnosed in the same way; the file is probably misnamed or is not in the same directory as OPSPRI.EXE.

Messages that look the same as the above message but that contain the name of a curve file or a scenario file, should be dealt with as follows: Compare the file name in CROSSREF.DAT with the file names for curve files (or scenario files) in the directory specified in CROSSREF.DAT. If you cannot find a match, either the curve file (or scenario file) is not in the directory specified in CROSSREF.DAT or the name of that curve file (scenario file) is not represented in the same way in both locations. For example, the message

> Cannot open file F15.CRV. Please check to see if F15.CRV is in your input file directory.

will appear if the 1000th TFS is paired with the curve file and path D:\TAC\LANTCOM\F15\F15.CRV in the file CROSSREF.DAT, but the curve file name in the directory D:\TAC\LANTCOM\F15 appears as F015.CRV. The same error message will appear if the curve file F15.CRV is missing or if it exists but not in the directory D:\TAC\LANTCOM\F15.

FILE READING ERRORS

A message that says,

> Error occurred in attempting to read the file PRIORITY.DAT.

probably is caused by attempting to create the disk data base using OPSPRI Version 1.1 with the PRIORITY.DAT file for Version 1.0 or by having rebooted your computer in the middle of a previous data base creation. Save the old file under a different name if you wish to keep it, replace it with a new file in the correct format, run OPSPRI, and select *YES* on the first menu – re-create the data base. This will overwrite old data base files with files in the correct format. The same error message with CROSSREF.DAT instead of PRIORITY.DAT has a similar solution – replace the old CROSSREF.DAT with a new one in the correct format, and re-create the data base. A similar solution will also work if the file name is a list file (e.g., MDLIST.DAT), a curve file (e.g., F016.CRV), or a scenario file (e.g., F016.SCN); in each case, replace the offending file by one in the correct format, and re-create the disk data base.

If you have selected *NO* (do not re-create the disk data base), you may see the same error message but with one of the following file names:

- UNIT.DAT
- PRIGP.DAT
- COMMAND.DAT
- MD.DAT
- PRGPTHTR.DAT
- THTR.DAT
- MDKITS.DAT
- CMNDKITS.DAT

- PRGPKITS.DAT
- PRTHKITS.DAT
- THTRKITS.DAT
- AGGNUMBS.DAT.

Again, the most probable causes are that the file named in the error message is an OPSPRI Version 1.0 file that you are attempting to read with OPSPRI Version 1.1 or that you rebooted your computer in the midst of data base creation. To resolve this problem, rerun OPSPRI and choose *YES* (re-create the data base). That procedure will overwrite the files with new data base files in the correct format.

APPENDIX A

INPUT FILE FORMATS

All Operational Priorities (OPSPRI) model input files, except for graphics files, are ordinary text files. OPSPRI input files either are used directly or are produced from other files by preprocessor programs. The first section of this appendix describes the format of directly used files; the second section describes the input files for the preprocessors and the OPSPRI input files they produce.

DIRECTLY USED INPUT FILES

Directly used OPSPRI input files are either list files, scenario files, or graphics files. No preprocessor programs are used with these files.

Mission Design List File

The mission design (MD) list file is named MDLIST.DAT. Its format is as follows:

- *Line 1:* Header – may contain any message; ignored by OPSPRI.
- *Line n, n > 1:* MD name, 1 to 5 characters, left-justified.

Major Command List File

The major command (MAJCOM) list file is named CMNDLIST.DAT. Its format is as follows:

- *Line 1:* Header – may contain any message; ignored by OPSPRI.
- *Line n, n > 1:* MAJCOM name, 1 to 8 characters, left-justified.

Theater List File

The theater list file is named THTRLIST.DAT. Its format is as follows:

- *Line 1:* Header – may contain any message; ignored by OPSPRI.
- *Line n, n > 1:* Theater name, 1 to 9 characters, left-justified.

Priority Group List File

The priority group list file is named PRGPLIST.DAT. Its format is as follows:

- *Line 1:* Header – may contain any message; ignored by OPSPRI.
- *Line n, n > 1:* Priority group name, 1 to 12 characters, left-justified.

Scenario Files

Scenario files are furnished by the Requirements/Execution Availability Logistics Module (REALM) of the Weapon System Management Information System (WSMIS). They should be given file names with the extension *.SCN*; for example, *F016.SCN*. Their formats are as follows:

- *Line 1:* Columns 1–3 contain the number of days in the scenario, right-justified.
- *Line n, 1 < n < days + 3:* Columns 1–8 contain flying hours for Day n – 2, with the decimal point in Column 6.

Graphics Files

Graphics files – in Turbo Pascal 5.5 format – are not accessible to the user. These files are

- ATT.BGI
- CGA.BGI
- EGAVGA.BGI
- HERC.BGI
- IBM8514.BGI
- PC3270.BGI
- SANS.CHR
- TRIP.CHR.

PREPROCESSOR INPUT FILES AND ASSOCIATED OPSPRI INPUT FILES

OPSLOG.DAT File

The OPSLOG.DAT file is an input file for the preprocessor OPSMERGE. It has the following format:

- *Lines 1 –2:* Header lines – may contain messages; ignored by OPSMERGE.
- *Line n, n > 1:*
  - Columns 1 – 12: Priority group, left-justified
  - Columns 13 – 14: Blanks
  - Columns 15 – 23: Theater, left-justified
  - Columns 24 – 25: Blanks
  - Columns 26 – 28: Support Level A, right-justified integer
  - Column 29: Blank
  - Columns 30 – 32: Support Level B, right-justified integer
  - Column 33: Blank
  - Columns 34 – 36: Support Level C, right-justified
  - Column 37: Blank
  - Columns 38 – 125: Support Levels D – Y
  - Columns 126 – 128: Support Level Z, right-justified integer.

[*Note:* Support levels do not need to go all the way to Z. If support levels stop before Level Z, the rest of the line must be left blank, and every line must have the same number of support levels.]

WMP3.DAT File

The WMP3.DAT file is an input file to the preprocessor OPSMERGE. It has the following format:

- *Lines 1 –2:* Header lines – may contain messages; ignored by OPSMERGE.
- *Line n, n > 1:*
  - Columns 1 – 13: Unit name, left-justified

- Columns 14 – 15: Blanks
- Columns 16 – 18: SUB – squadron/wing detachment (subset of a unit – characters)
- Columns 19 – 20: Blanks
- Columns 21 – 29: Theater, left-justified
- Columns 30 – 31: Blanks
- Columns 32 – 43: Priority group, left-justified
- Columns 44 – 45: Blanks
- Columns 46 – 53: MAJCOM, left-justified
- Columns 54 – 55: Blanks
- Columns 56 – 61: MDS (MD series), right-justified
- Columns 62 – 63: Blanks
- Columns 64 – 67: Sortie length, right-justified
- Columns 68 – 69: Blanks
- Columns 70 – 72: primary aircraft authorized (PAA).

PRIORITY.DAT File

The PRIORITY.DAT file is an OPSPRI input file produced by the preprocessor OPSMERGE from the files OPSLOG.DAT and WMP3.DAT. It has the following format:

- *Line 1:* Contains this message – *Unit-Level Ops Priorities Matrix.*
- *Line 2:* Contains column labels.
- *Line 3:*
  - Columns 1 – 10: Contain this message – *NSUPPORTS=.*
  - Column 11: Blank
  - Columns 12 – 13: Two-digit number of support levels.

- *Line n, n > 3:*
  - Columns 1–16: Unit name and SUB (squadron/wing detachment – subset of unit), left-justified
  - Columns 17–18: Blanks
  - Columns 19–23: MD, right-justified
  - Columns 24–25: Blanks
  - Columns 26–37: Priority group, left-justified
  - Column 38: Blank
  - Columns 39–47: Theater, left-justified
  - Column 48: Blank
  - Columns 49–56: MAJCOM, left-justified
  - Columns 57–58: Blank
  - Columns 59–63: Sortie length
  - Column 64: Blank
  - Columns 65–67: Support Level A – three-digit integer, right-justified
  - Column 68: Blank
  - Columns 69–71: Support Level B – three-digit integer, right-justified
  - Column 72: Blank
  - Columns 73–75: Support Level C – three-digit integer, right-justified
  - Column 76: Blank
  - Columns 77–164: Support Levels D–Y
  - Columns 165–167: Support Level Z.

[*Note:* Support levels need not go all the way to Level Z, but the number of support levels must be the same on every line and must equal the number specified on Line 3.]

UNITKIT.DAT File

The UNITKIT.DAT file is an input file to the preprocessor UNITOCRV. It has the following format:

- *Lines 1 and 2:* Header lines – may contain messages; ignored by UNITOCRV.
- *Line n, n > 2:*
  - Columns 1 – 13: Unit name, left-justified
  - Columns 14 – 15: Blanks
  - Columns 16 – 18: SUB – squadron/wing detachment (subset of a unit)
  - Columns 19 – 20: Blanks
  - Columns 21 – 32: First 12 characters of kit serial number (KSN).

KITFILE.DAT File

The KITFILE.DAT file is an input file to the preprocessor UNITOCRV. It has the following format:

- *Lines 1 and 2:* Header lines – may contain messages; ignored by UNITOCRV.
- *Line n, n > 2:*
  - Columns 1 – 12: KSN, left-justified
  - Columns 13 – 20: Blanks
  - Columns 21 – 60: REALM availability curve file name and path name for primary day of scenario, left-justified. File name extension must be *.CR1*.
  - Columns 61 – 62: Blanks
  - Columns 63 – 103: Scenario file name and path name, left-justified.

CROSSREF.DAT File

The CROSSREF.DAT file is an OPSPRI input file produced by the preprocessor UNITOCRV from the files UNITKIT.DAT and KITFILE.DAT. It has the following format:

- *Line 1:* Contains this message – *Unit-Curve File Name/Scenario File Name Cross-Reference File.*
- *Line 2:* Contains column labels.
- *Line n, n > 2:*
  - Columns 1 – 16: Unit name and SUB (squadron/wing detachment), left-justified
  - Columns 17 – 19: Blanks
  - Columns 20 – 59: Relative availability curve path/file name. File name extension must be *.CRV*.
  - Columns 60 – 99: Scenario path/file name. File name extension must be *.SCN*.

CRVFILES.DAT File

The CRVFILES.DAT file is an input file to the preprocessor MAKECRVS. It has the following format:

- *Line 1:* Header line – contains message; not used by MAKECRVS.
- *Line n, n > 1:*
  - Columns 1 – 40: Contain REALM curve path/file name for primary day, left-justified. Must end with extension *.CR1*.
  - Columns 41 – 80 (optional): Contain REALM curve path/file name for secondary day, left-justified. Must end with extension *.CR2*.
  - Columns 81 – 83: Contain primary day integer, right-justified.
  - Column 84: Blank.
  - Columns 85 – 87 (optional): Contain secondary day integer, right-justified.

WSMIS/REALM Curve Files

These curve files are input files to the preprocessor MAKECRVS. They have the following format:

- *Line 1:* First 12 characters of a KSN, left-justified, followed by spaces, integer PAA, spaces, and real number direct support objective (DSO) classic (aircraft down). The number of spaces separating the entries does not matter.
- *Line 2:* Contains message – *Will be copied to relative availability curve file by MAKECRVS.*
- *Line n, 2 < n < last – 1:*
  - Columns 1 – 3: Blanks
  - Columns 4 – 10: Expected not mission capable–supply (ENMCS) – real number with decimal point in Column 7
  - Columns 11 – 26: Ignored
  - Columns 27 – 36: Cost – real number with decimal point in Column 36
  - Columns 37 – 80: Ignored.
- *Line last – 1:*
  - Columns 1 – 5: Blanks
  - Columns 6 – 10: Contain 0.000
  - Columns 11 – 26: Ignored
  - Columns 27 – 34: Blanks
  - Columns 35 – 36: Contain 0.0
  - Columns 37 – 80: Ignored.
- *Last Line:* Ignored.

Relative Availability Curve Files

Each of these curve files is produced from either one or two WSMIS/REALM aircraft availability curve files. For a description of the process, see Appendix C. The file name extension must be *.CRV*. Each file is a text file with the following format:

- *Line 1:* Contains the message – *KITID/PAA/DSOHI/DSOLO/DAYHI/DAYLO:.*
- *Line 2:* First 12 characters of a KSN, spaces, PAA, spaces, primary day curve DSO (aircraft up), spaces, secondary day curve DSO, spaces, primary day, spaces, secondary day.
- *Line 3:* Contains the message – *Primary Day Curve:.*
- *Line 4:* Contains header from primary day curve.
- *Line 5:* Contains the message – *Secondary Day Curve:.*
- *Line 6:* Contains header from the secondary day curve.
- *Line 7:* Contains the message – *COST/RELAVAIL/HIAVAIL/LOAVAIL:.*
- *Line n, n > 7:*
  - Columns 1 – 3: Blanks
  - Columns 4 – 14: Cost
  - Columns 15 – 17: Blanks
  - Columns 18 – 24: Relative availability
  - Columns 25 – 27: Blanks
  - Columns 28 – 34: Primary day curve relative availability
  - Columns 35 – 37: Blanks
  - Columns 38 – 44: Secondary day curve relative availability.

APPENDIX B

OUTPUT FILE FORMAT

The Operational Priorities (OPSPRI) model produces one output file, called UNIT.OUT that is returned to the Requirements/Execution Availability Logistics Module of the Weapon System Management Information System for war readiness spares kit/base level self-sufficiency spares budget allocation. It has the following format:

- *Line 1:* Header – Contains message.
- *Line n, n > 1:*
  - Columns 1 – 16: Contain unit name and unit subset designator, if any, left-justified
  - Columns 17 – 18: Blanks
  - Columns 19 – 29: Dollars allocated to unit; decimal point in Column 29, right-justified
  - Columns 30 – 31: Blanks
  - Columns 32 – 34: Three-digit integer availability, right-justified.

APPENDIX C

RELATIVE AVAILABILITY

This appendix presents an explanation of the concept of relative availability and describes how the preprocessor MAKECRVS creates a relative availability curve from two availability curves. In the discussion below, PAA means primary aircraft authorized, ENMCS means expected not mission capable–supply, and DSO means direct support objective.

For a single aircraft availability curve, we define its associated relative availability curve as follows:

Each point on the original curve is of the form

$$(\textit{Cost}, \textit{Availability}),$$

where

$$\textit{Availability} = 100 \times (\text{PAA} - \text{ENMCS})/\text{PAA}.$$

The new curve consists of the points

$$(\textit{Cost}, \textit{Relative Availability})$$

where

$$\textit{Relative Availability} = 100 \times ((\text{PAA} - \text{ENMCS})/\text{DSO}).$$

where DSO is the required number of aircraft up. Hence, relative availability is "availability relative to the DSO." Relative availability can exceed 100 percent.

We now define a single relative availability curve made from two relative availability curves, one for a primary day and one for a secondary day.[1] In the Requirements/Execution Availability Logistics Module (REALM) of the Weapon System Management Information System (WSMIS), an availability curve is produced for the primary day (usually Day 7) first. The curve for the secondary day is

[1]The *primary* day and the *secondary* day are defined in Chapter 7 of the main text in the section dealing with "Generating Reports."

generated after adding in the primary day's requirement to the assets for the secondary day. Thus the cost field in the secondary day's curve represents cost over and above the primary day's requirement. This process motivates our definition of a relative availability curve made from curves for two distinct days. For brevity, we refer to the first curve as the primary day curve, and the second as the secondary day curve. The process we describe is used in MAKECRVS.

First we consider the first point on the secondary day curve, where cost is a minimum. If the minimum cost on the secondary day curve is not zero, we must generate an extra point with zero cost and estimated availability "Estavail", computed by extrapolating as follows:

(cost(1), avail(1))	=	first point on secondary day curve
(cost(2), avail(2))	=	second point on secondary day curve
slope	=	[log(avail(2)) – log(avail(1))]/[cost(2) – cost(1)]
Estavail	=	exp{ log[avail(1)] – slope × cost(1)}.

We use linear extrapolation on the log availability curve, followed by exponentiation, to avoid the possibility of Estavail being negative.

Now we have a secondary day curve with a new first point having cost = 0. Call the new first point (cost(1), avail(1)). The old first point becomes the new second point (cost(2), avail(2)), the old second point becomes the new third point (cost(3), avail(3)), and so on.

Let

N1	=	number of points on primary day curve
N2	=	number of points on secondary day curve
Cost1(n)	=	cost for nth point on primary day curve
Cost2(n)	=	cost for nth point on secondary day curve
Costrel(n)	=	cost for nth point on relative availability curve
Avail1(n)	=	availability for nth point on primary day curve
Avail2(n)	=	availability for nth point on secondary day curve
Availrel (n)	=	availability for nth point on secondary day curve.

For n = N1, we set

$$Costrel(n) = Cost1(n)$$

$$Availrel(n) = \min\{Avail1(n), Avail2(1)\}.$$

For N1 < n < N1 + N2, we set

$$Costrel(n) = Cost2(N1) + Cost2(n - N1 + 1)$$

$$Availrel(n) = \min\{Avail1(N1)\}, Avail2(n - N1 + 1)\}.$$

For 0 < n < N1, we first estimate the availability obtained on the secondary day curve by spending Cost1(n) dollars on the primary day curve:

$$Estavail2(n) = Avail1(n) \times [Avail2(1)/Avail1(N1)].$$

We then define the points on the relative availability curve for 0 < n < N1 by:

$$Costrel(n) = Cost1(n)$$

$$Availrel(n) = \min\{Avail1(n), Estavail2(n)\}.$$

Note that this way of defining relative availability assumes that even when we have spent enough to reach Avail1(N1) = 100 on the primary day, the relative availability on the secondary day will be the same as what we would get by spending zero dollars on the secondary day curve.

In reality, the availability on the secondary day will be higher than this, because spares purchased to improve availability on the primary day will also improve availability on the secondary day. In this sense, our definition of a single relative availability curve, made from curves for two distinct days, is conservative.

APPENDIX D

GLOSSARY

AFLC	=	Air Force Logistics Command
ALASKCOM	=	Alaskan Command
BLSS	=	base level self-sufficiency spares
CENTCOM	=	Central Command
CGA	=	color graphics adapter
DOS	=	disk operating system
DSO	=	direct support objective
EGA	=	enhanced graphics adapter
ENMCS	=	expected not mission capable–supply
EUCOM	=	European Command
IBM	=	International Business Machines
KSN	=	kit serial number
LANTCOM	=	Atlantic Command
LMI	=	Logistics Management Institute
MAJCOM	=	major command
MD	=	mission design
MDS	=	mission design series
NMCS	=	not mission capable–supply
OPS	=	operational
Ops/Log	=	Operations Logistics
OPSPRI	=	Operational Priorities (model)
PAA	=	primary aircraft authorized

PACOM	=	Pacific Command
PC	=	personal computer
RA	=	relative availability
RAM	=	random access memory
REALM	=	Requirements/Execution Availability Logistics Module
SOUTHCOM	=	Southern Command
SPM	=	system program manager
TAC	=	Tactical Air Command
TFS	=	Tactical Fighter Squadron
TRANSCOM	=	Transportation Command
VGA	=	video graphics array
WMP	=	War and Mobilization Plan
WRSK	=	war readiness spares kits
WSMIS	=	Weapon System Management Information System

REPORT DOCUMENTATION PAGE

Form Approved
OPM No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 91	Final

4. TITLE AND SUBTITLE
Operational Priorities Model (OPSPRI) Version 1.1: Users Manual

5. FUNDING NUMBERS
C MDA903-90-C-0006
PE 0902198D

6. AUTHOR(S)
Tovey C. Bachman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Logistics Management Institute
6400 Goldsboro Road
Bethesda, MD 20817-5886

8. PERFORMING ORGANIZATION REPORT NUMBER
LMI-AF001R1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Director of Supply
Headquarters, U.S. Air Force
The Pentagon, Room 4E260
Washington, DC 20330

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
A: Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This report describes the Operational Priorities (OPSPRI) model, which allocates funds for Air Force war readiness spares kits in accordance with operational priorities. It includes explanations of model installation, operation, and use in providing policy guidance and describes input file formats.

14. SUBJECT TERMS
OPSPRI, Operational Priorities model, war readiness spares kits (WRSK), base level self-sufficiency spares (BLSS), budget allocation, prepositioned, prestocked

15. NUMBER OF PAGES
86

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	18. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL